Don't Believe Everything you Hear: Combining Audio and Visual Cues for Deepfake Detection

Kyra Mozley

CDT Summer Project

August 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This project aims to use deep learning techniques to detect whether or not a video is a deepfake. To achieve this goal, we extract relevant visual and audio features on a suitable dataset and implement a hybrid deep neural network to perform classification.

1.1 Motivation

Advancements in computer graphics, computer vision, and machine learning have made it easier than ever before to synthesise highly realistic audio, image, and video. Content creators can take advantage of these developments and, with enough data, can generate fake content. Deepfakes, a combination of the words 'deep learning' and 'fake', refer to content created by manipulating media, often an actor swapping their face to impersonate someone else, to create fake footage that seems authentic.[4]

Unfortunately, deepfakes offer the potential to cause much damage to society. Most people trust footage circulated on social media[5] since historically we rely on video footage as evidence of events. However, as deepfake technology improves, and it becomes almost impossible for individuals to distinguish manipulated content from genuine, it opens the door to the possibility of using this technology for spreading disinformation. The development of deepfakes will lead to new types of deception, extortion, or bullying.[6] Current concerns relate to creating false pornographic content (revenge porn)[7], the threat to election integrity[8] and the ability to conduct fraud[9].

Nevertheless, we must state that there has also been positive usage of this technology. For example, it has proved revolutionary for the multimedia industry, allowing for the inclusion of deceased individuals in their content or to achieve high-quality visual effects cheaper and easier than ever before.[7] However, the malicious use of deepfakes will likely dominate the valid ones. Therefore to combat the potential harm, we must create automatic methods to detect deepfakes so that such videos can be flagged online to make the viewer aware that this is not genuine content.

1.2 Project Description

This project explores the link between emotion conveyed in the physical aspects of the individual (e.g. their facial expressions or head position) with the emotion in their speech to classify a video as real or fake.

Inspired by *Emotions Don't Lie*[10] we argue that there must exist a relationship between the audio-visual modalities of an individual. More specifically, we hypothesise that emotions extracted from visual cues such as facial expressions should correlate with the emotion conveyed in the subjects' speech. Deepfake videos have undergone manipulation, often altering the mouth region to match the impersonators' identity, speech, or expression.[11] Therefore, the creation of deepfakes should occasionally result in the disruption of the audio-visual relationship of expression, and this is what we aim to exploit to aid detection of such videos. However, unlike much of the research that approaches this problem from an emotion recognition angle, we do not explicitly map the selected features to one of several discrete emotions. Instead, we argue that using discrete emotions may hinder performance since human emotion is not discrete, plus emotion can be faked[12], or there may be more than one perceived emotion in the same utterance.[13] Thus, leaving the extracted features in their raw form provides more information to the classifier, and therefore should improve performance.

Using reasoning grounded in psychology and affective computing, we select relevant features that provide rich information regarding sentiment. For example, it is well understood in the literature that the spoken word only accounts for a fraction of how we communicate and that non-verbal actions provide more information about an individual's feelings; this is known as Albert Mehrabian's 7/38/55 rule of communication.[14] With 7% of expression information provided through words, 38% through tone of speech, and 55% through facial expression and body language. Therefore, by combining audio-visual modalities, we can create richer algorithms than others that use either unimodal approaches or multimodal models whose modalities are exclusively from the visual stream.

1.3 Main Contribution

We present a novel approach to deepfake detection that extracts audio-visual features inspired from emotion recognition literature to detect whether a video is real or fake. We are the first multimodal approach whose reasoning is based in emotion recognition not to map the audio-visual modality to a discrete emotion. Our learning method uses a hybrid network consisting of convolutional neural networks (CNNs) and bidirectional long short term memory (Bi-LSTMs) to achieve a final AUC of 96.2% on the Deepfake Detection Challenge dataset[15], improving on existing multimodal approaches. Additionally, we explored different fusion techniques to combine the audio and visual modalities. Whatsmore, we show the ability of our model to generalise; we test on a different dataset (DF-TIMIT), for which had not been seen in training, and achieve a near-perfect AUC score of 99.9%.

1.4 Overview of the Report

This report is split into seven different chapters:

- *Chapter One: Introduction:* In this chapter, we have presented the motivation behind the need for deepfake detection, as well as the project description.
- *Chapter Two: Background:* Covers the background theory needed for this project, focusing on digital signal processing knowledge, relevant deep learning definitions, and deepfake generation.
- *Chapter Three: Previous Works:* We present an overview of affective computing literature alongside the current approaches to deepfake detection and discuss the state-of-the-art methods.
- *Chapter Four: Our Approach:* Discusses the features and method we have chosen to take and the reasoning behind these choices.
- *Chapter Five: Implementation:* Provides details of the steps taken to extract the chosen features and implement the deep learning models.
- *Chapter Six: Results and Discussion:* Considers the results of our detection architecture and works towards building the final model. It also discusses our performance on other datasets, the limitations of our method, and which features are of most importance for classification.
- *Chapter Seven: Conclusion:* Reviews the work completed, giving final remarks on the success and possible future work.

Chapter 2

Background

This chapter outlines the relevant background knowledge needed for this project, covering how to process audio signals, deep learning techniques, and how to evaluate them. Lastly, we turn to look at how deepfakes are generated.

2.1 Digital Signal Processing

Digital signal processing (DSP) involves manipulating signals that originated in the analogue world, such as audio signals. So we must apply appropriate processing to ensure we do not lose information from the original signal. We now discuss the relevant DSP techniques that were required for this project.

2.1.1 Pre-Processing

Audio pre-processing refers to all the operations performed on samples of a signal before we extract our desired features. It is essential in systems where background noise or silence is undesirable, for example, or to normalise utterances since different recording environments may result in different energy levels.[16]

The first step we apply to our extracted audio signal is **pre-emphasis**, a filtering technique that boosts the high frequencies of a speech signal. Which, in turn, flattens the spectrum, reducing the height of the dynamic spectral range.[17] It is calculated as

$$y(t) = x(t) - \alpha x(t-1)$$

for an input signal $x(t)$ where α is a coefficient that controls the pre-emphasis filter, often set to 0.97 for speech recognition. Figure 2.1 shows the result of applying pre-emphasis, and the flattened spectrum as a result.

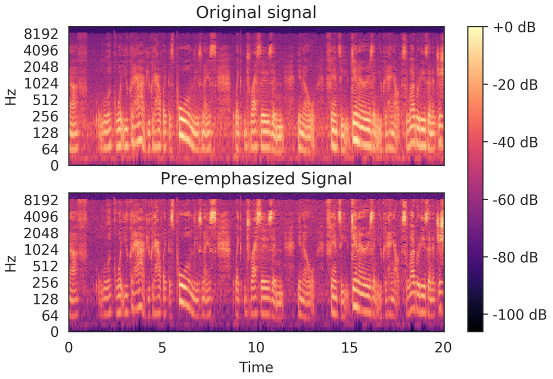

Figure 2.1: Two spectrograms, showing the effect of pre-emphasis on a signal. The signal with pre-emphasis applied has had its lower frequencies reduced, overall flattening the spectrum.

Next, we must divide the given continuous speech signal into segments of fixed length called frames. While the properties of the audio signal will vary over the given audio clip, the signal can be assumed stationary for a short period (20-50ms). Thus, **framing** the signal allows us to extract local features.[17] Moreover, in order to smooth the signal and ensure information between frames is preserved, overlapped frames are used. Typically, in speech processing, a frame has a 25ms duration with a 10ms overlap between consecutive frames.

When we perform Fourier calculations on the signal, the discrete Fourier transform (DFT) will assume a frame contains one period of a periodic sequence, but this is unlikely to be the case; instead, there are likely sharp boundaries at the edges of these sampled frames.

When these discontinuities are present in the DFT calculation, it results in a phenomenon known as spectral leakage, where the spectrum returned would not be the actual spectrum of the original signal but a smeared one. Therefore, we must apply a function that smooths the input, bringing the amplitude to almost zero at the edges.[18] Ergo, the last step we perform in pre-processing is windowing. We multiply all frames with a **windowing function**, which is often maximal in the middle and tapers off at the ends. In speech processing, the signal is often multiplied by the Hamming window, shown in Figure 2.2, and is given as:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$

where $0 \leq n \leq N-1$ if N is the window length in samples.[16]

2.1.2 Mel-Frequency Cepstrum

The audio signal is hard to process directly as a potential feature input, so we must extract audio features that represent properties of the original signal, reducing the volume of data. **Mel-frequency cepstrum coefficients** (MFCCs) are commonly used in speech processing and offer a cepstral based signal representation.

The details of how to calculate the MFCCs are involved and out of the scope of this report. However, what is important to note is that it operates in the frequency domain, using the Mel-scale to distribute the centre frequencies of each filter. The Mel-scale more closely approximates the human auditory response than typical linear-spaced frequencies

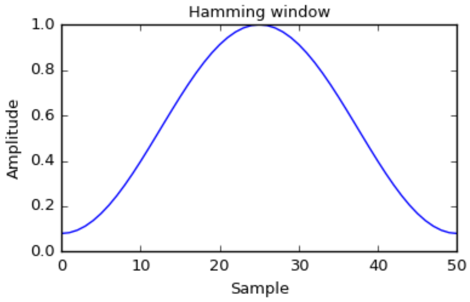

Figure 2.2: A graph to show the Hamming Window function.

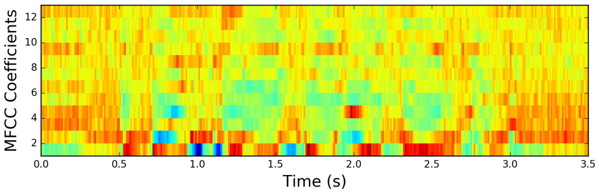

Figure 2.3: A spectrogram of the first 13 MFCCs extracted from an audio signal.

in other spectrums.[19] Overall, the MFCCs are a set of coefficients that represent speech signals, and the first 8 to 13 cepstral coefficients are commonly used since they contain the majority of the signal information.[17]

2.2 Relevant Deep Learning Knowledge

To understand how deep fakes are created and how we can utilise deep learning to detect them, we first discuss the relevant machine learning theory required for this report.

2.2.1 Deep Learning Methods

We provide an overview of the different deep learning models that are discussed later.

Multilayer Perceptrons

Artificial neural networks are designed to mimic the structure of the human brain. A multilayer perceptron (MLP) consists of layers of nodes (or neurons.) There are no connections between these nodes within a layer, but each node is fully connected to nodes in adjacent layers. Each of these connections has a weight associated with them. This weight measures the degree of correlation between the activity of these nodes.[20] During training, we provide a series of input vectors and the associated output vectors. Then, the weights in the network are continuously adjusted to give the desired outcome - in our case, real or fake.[21] The MLP was the first implementation of an artificial neural network[22] and is still one of the most utilised models today.[23] Note that some libraries, and our implementation, refer to an MLP as a dense layer.

Convolutional Neural Networks

A particular type of artificial neural network is a Convolutional Neural Network, or CNN. It can learn highly abstracted features of objects, inspired by the organisation of the visual cortex in the human brain, and is therefore often utilised for image recognition. [24] It also reduces the chances of overfitting and improves generalisation, which is why it is extensively used over classical neural networks. A CNN can learn local responses from temporal or spatial data but cannot learn sequential correlations.[25]

Long Short-Term Memory

A Long Short-Term Memory (LSTMs) is a particular type of recurrent neural network (RNN) that can learn both short-term and long-term dependencies in time-series problems.[25] LSTMs consist of layers of a set of recurrently connected blocks - their memory blocks. Each block contains one or more connected memory cells plus the input, output, and forget gates.[26] They take advantage of the memory units to maintain the state over time and is considered state of the art for sequence data.[27]

One variant that improves the performance and generalisation power of LSTMs is a bidirectional LSTM. A BiLSTM consists of two LSTMs that process the input sequence in different directions. One reads the sequence in its input order and the other in reverse, it then merges these representations.[28] By processing the input in two directions, a BiLSTM catches time series patterns that may elude unidirectional LSTMs.[29]

Generative Adversarial Networks

A Generative Adversarial Network (GAN) consists of two parts; a generator and a discriminator. During training, we provide a dataset that consists of real examples, and the generator aims to produce data points that are similar to the given actual data. Meanwhile, the discriminator aims to distinguish images generated by the generator from real examples. We train the discriminator to maximise the probability that it assigns correct labels to both the real data and that generated by the generator. Simultaneously, the generator is trained to minimise the probability that the discriminator detects its output as fake. After training, the generator produces convincingly fake content, whilst the discriminator's ability to detect generated data from real will also be increased.[30]

AutoEncoders

An autoencoder is a specific neural network designed to encode the input into a latent, compressed representation. Subsequently, it learns how to decode it back from this compressed form, where the reconstructed input is as similar as possible to the original input.[31] Figure 2.4 illustrates the autoencoder model. They differ from GANs since the autoencoder learns how to provide meaningful representations of data, whilst a GAN uses an adversarial feedback loop to learn how to generate information that looks real.

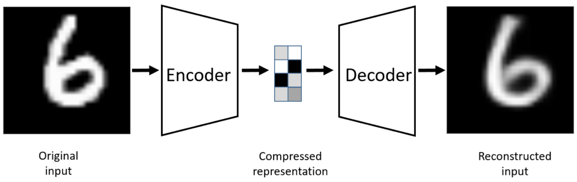

Figure 2.4: An autoencoder example. The input image is encoded to a compressed representation and then decoded. Image courtesy of [1]

	Predicted Fake	Predicted Real
Truly Fake	True Positive (TP)	False Negative (FN)
Truly Real	False Positive (FP)	True Negative (TN)

Table 2.1: The structure of a confusion matrix for the deepfake detection problem.

2.2.2 Evaluation Metrics

We require evaluation metrics to judge the quality of the learning algorithm we have presented. We now explore the relevant metrics for this report.

A **confusion matrix** allows us to visualise the performance of the classification algorithm; it is a table whose results are divided into actual and predicted classes.[32] Table 2.1 shows a confusion matrix for a binary classification algorithm.

Accuracy is the number correct classifications, divided by the total number of classifications.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Although this may seem like a simple metric, it can fail to describe the true performance if we have highly imbalanced classes.[32]

We define **precision** as the ratio of the number of class predictions that truly were that class, over the number that was predicted that class.[33]

$$\text{Precision} = \frac{TP}{TP + FP}$$

A low precision indicates that we have a large number of false positives in our results. Classifying genuine content as deepfake too often, the false-positive case, can render the system unusable in the wild due to too many false alarms. The **false positive rate** is the measure of the false alarm rate and is given as

$$FPR = \frac{FP}{FP + TN}$$

In our case, it measures the number of real videos that have been labelled as deepfakes.

Recall, also called **true positive rate**, is the ratio of the number of class predictions that truly were that class, over the number of true occurrences of that class.

$$\text{Recall (or TPR)} = \frac{TP}{TP + FN}$$

A low recall indicates that we have a large number of false negatives in our results. We wish to avoid classifying deepfake footage as real; this is the false-negative case and would lead to undetected manipulated content, meaning the detector fails to do the exact task it was created for. Overall, we want to achieve both high results for precision and recall values.

A receiver operating characteristics (**ROC**) graph is a common visualisation technique. The true positive rate is plotted on the Y-axis, and the false positive rate on the X. This graph demonstrates the tradeoff between true and false positives. Figure 2.5 provides an example of a ROC graph. There are some notable points on this graph; (0,1) represents perfect classification, whilst the line from (0,0) to (1,1) occurs if the binary classifier was guessing the result - hence it is called the no skill line. Overall, one line is better than another the further north-west the line is, and if it falls below the no skill line, it is considered is worse than randomly guessing.[34]

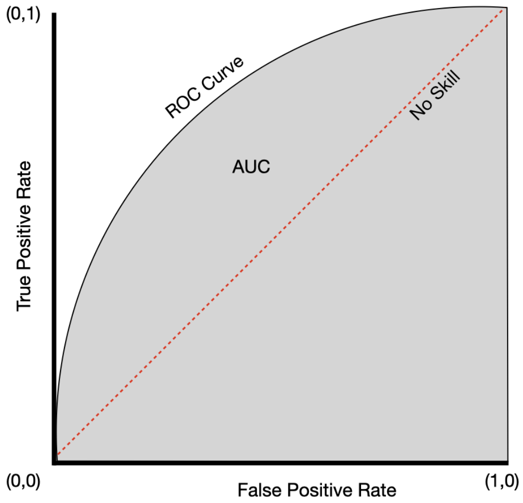

Figure 2.5: An example of a basic receiver operating characteristcs (ROC) graph. The shaded grey area represents the area under the ROC curve, or the AUC.

The last metric we look at is the area under the ROC curve (**AUC**). It is a popular metric to assess the performance of a binary classifier compared to randomness (which achieves an AUC score of 0.5), a model whose predictions are all correct will have an AUC score of 1.0, and a model whose predictions are all incorrect will have a score of 0.0. Some researchers consider AUC to be a better metric than accuracy.[35]

In this report, we will provide both AUC and accuracy as a means to evaluate and compare performance.

2.3 Deepfake Generation

Deep learning has driven the advancement of deepfakes. We can generate manipulated content using machine learning techniques, typically autoencoders and GANs, trained on

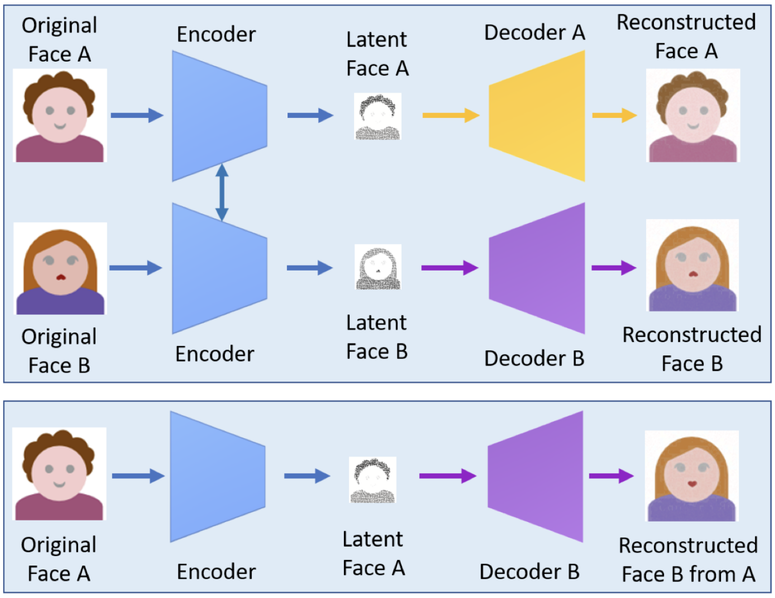

Figure 2.6: FakeApp deepfake generation. (Top) Training two encoder-decoder pairs, with a shared encoder. (Bottom) Image of person A is encoded with the shared encoder, but uses person B's decoder to generate the deepfake image. Image courtesy of [2]

an extensive dataset of footage from the target person. The first popular deepfake video surfaced in 2017 and contained a celebrities face that had been superimposed on a porn actor.[36] This footage had been created using the FakeApp network, which, pictured in Figure 2.6, uses an autoencoder to extract the hidden features of both individuals and a decoder to reconstruct the facial image. These two encoder-decoder pairs are individually trained on an image set, with the parameters of the encoder network shared between the two pairs. This encoder can learn the comparison between the two sets of faces. Then, to generate the target face, we pass the impersonators footage through the joint encoder but use the targets decoder to retrieve the deepfake video.[37]

Currently, there are three main types of deepfake videos. Firstly, **head puppetry** generates a video of a target person's whole head and upper shoulder from a video of a source person's head. The synthesised target behaves the same way as the source - as if the target were a puppet. Next, **Face Swap**, like in Figure 2.6, creates a video of the target by replacing their face on the imposters. Lastly, **Lip Syncing** creates manipulated video by only editing the area around the lips so that the target appears to say something different from the original video.[38]

Initially, one of the most significant challenges deepfake creation would have faced would be computational cost. However, recent advancements in computer graphics have reduced this barrier to entry, and almost anyone with some degree of programming skills can generate them using the available tools online.[2] Now, one of the most considerable challenges observed in deepfake creation is the requirement for a large number of training images of the subject.[2] It is this reason that most deepfake footage generated online is that of famous people since they have hours of media content online on sites such as YouTube.

2.4 Summary

In this chapter, we have covered the required background knowledge needed for the rest of this report. Starting with audio processing techniques and MFCCs, then covering the relevant deep learning techniques (CNNs, LSTM) and their evaluation metrics. Lastly, we look at how deepfakes are created to understand the motivation of the current detection methods in the next chapter.

Chapter 3

Previous Works

This chapter covers affective computing literature to develop a grounding in how emotions were traditionally detected in media. We then present an overview of the currently available datasets which are publicly available for deepfake detection. Finally, we turn to review unimodal and multimodal methods that are presently being employed for deepfake detection.

3.1 Affective Computing Literature

Before we can understand how to detect emotions in deepfake footage, we must first understand the current methods of detecting emotion in sound and video clips of genuine media. Therefore, we turn to affective computing, which is an area that relates to the detection or deliberate induction of emotion.[39] Here, we focus on how people have used technology for the detection of emotional states rather than invoking emotions in the user.

3.1.1 Facial Expression Recognition

Over the last twenty years, facial expression recognition has gained a lot of attention as a field given the possibility for its use in many applications such as human-computer interaction, interactive video, and much more.[40] The visual elements selected for this task include feature positions and shape changes caused by movements of facial characteristics and respective muscles during emotional expression.[41] Facial expression recognition can be complex since facial expressions may be subtle or momentary, or there may be multiple expressions presented at once.[42] Whatsmore, detecting these facial expressions can be difficult due to the occurrence of external noise such as inadequate illumination, occlusions or rotations.[43]

There are two main steps in facial expression recognition; feature extraction, and classification.[40] To perform facial feature extraction, we localise the face and obtain the feature points.

There are two approaches used within the literature that work on static images, appearance-based methods, operating over the entire image, or geometric-based, which extract landmark points based on geometric positions.[44] Appearance-based methods include applying Gabor wavelets or local binary patterns (LBP) to provide the texture and shape of the facial image. These appearance features comprise of micro-patterns that give information about facial expressions, yet they are difficult to generalise across the population due to the differences in facial structures.[45] The prefered approach to encode facial features is the **facial action coding system** (FACs), a geometric way to categorise facial actions based on the muscles that produce them.[43] Ekman and Friesen first proposed the FACs in 1978, which uses 44 anatomically based action units (AU) to describe a muscle area that relates to a given emotion, as seen in Figure 3.1. By tracking these AUs, we can reduce the effect of illumination, rotations, or any other noise that would degrade accuracy in any appearance-based methods.[45]

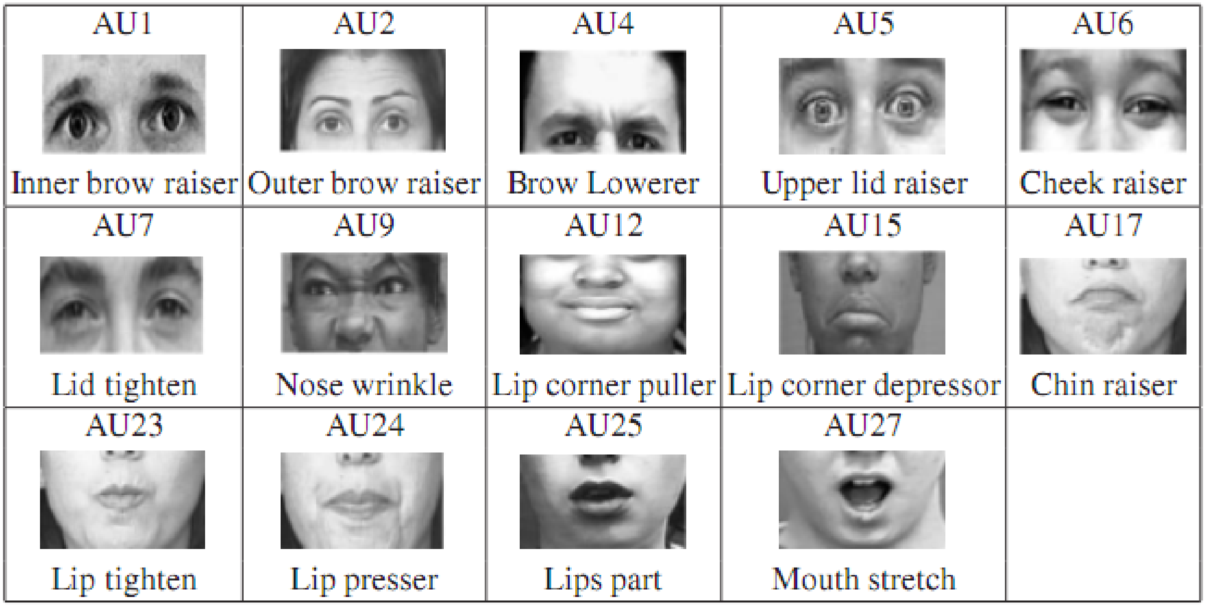

Figure 3.1: Shows examples of several facial action units from Ekman and Friesen's facial action coding system. Diagram from [3]

Once these features have been extracted using an appropriate method, we then perform classification using machine learning models such as a Support Vector Machine (SVM), or more recently, deep learning approaches such as CNNs.[44]

Whilst most literature focuses purely on facial movements as the visual aspect in emotion recognition, some might also include bodily postures, gestures or eye gaze. [46] Although the use of head pose and eye gaze is rare in emotion recognition, it has been shown that both of these features contain valuable emotional information, and therefore prove to be complementary features to facial expression.[47]

3.1.2 Speech Emotion Recognition

Speech emotion recognition is an equally as important branch of affective computing. Speech signals can carry rich and complex information about the speaker, such as their accent, gender, and emotion, in addition to the message.[48] However, speech recognition presents a challenging task since there is much variability across speakers, such as their accent, style, or rate of speech.[16] Moreover, there is no standard definition of emotion across the literature.[49] Academics have attempted to map emotions into six primary sentiments (anger, disgust, fear, joy, sadness, surprise) and the neutral label.[16] However, genuine emotions are not discrete, there may be more than one perceived emotion in a given utterance, and emotion portrayal generally depends on the speaker.

The typical pipeline for speaker emotion recognition consists of the following steps: data collection, signal pre-processing and segmentation, acoustic feature extraction, and lastly, classification.[48] In the past statistical learning methods such as SVMs or HMMs[16] were used to perform classification, but more recent papers have turned to use deep learning techniques, such as RNNs and CNNs.[48]

There exist a variety of different features that are available to use for speaker emotion recognition. These can be categorised into either time domain, frequency domain, prosodic, or short term spectral features.[48] Time domain features include the statistical properties of the speech signal and how they are dispersed, such as the signals mean, variance, and zero-cross rate. Frequency domain features represent the dispersal of signal energy; features include energy, spectral flux, and entropy. Prosodic features are long-term features that describe variations in energy (intensity), pitch, rhythm, and stress. Finally, we can use short term spectral features to extract information from frames. These low-level descriptors play an important role in the literature on speech emotion recognition.[50] Mel frequency cepstral coefficients (MFCCs) and linear prediction coefficients (LPC) are the most commonly used features in this area. Lower order MFCC features have been recognised to convey phonetic information, whilst higher-order cepstral coefficients carry non-speech information.[49]

3.1.3 Multimodal Emotion Recognition

Although initial human emotion recognition from video footage has been mostly unimodal and focuses on facial expressions, psychology research denotes the necessity of considering other cues to allow for more accurate predictions.[12] Emotion is expressed through our speech, facial expression, and body gestures, as well as through physiological changes such as heart rate, sweating degree etc.[40] Therefore, integrating multiple modalities can provide richer information than unimodal methods, leading to better results.

In [51] they demonstrated there exists a sufficient correlation between audio and visual features and that by considering the audio-visual world, the features are more robust to corruption from noise. Similarly, [42] found that audio modalities provided useful complementary information in addition to their visual modalities, leading to better performance for facial expressions in the wild. Lastly, [12] also praises multimodal features for emotion recognition due to the richer information the different cues provide, alongside the increased robustness to noise in the event one channel is corrupt or not present. Their feature modalities consisted of word embeddings, MFCCs and glottal source parameters for audio, and facial landmarks and AUs for the visual component. However, they raise the challenge of knowing how and when to combine the features. Most papers use additive combinations, but this assumes that every modality is always potentially useful, which may not be the case.

Overall, exploiting the potential of multiple modalities can improve the performance of emotion recognition classification.

3.2 Deepfake Detection

Due to the rise in the creation of deepfakes, and the concerns their use poses, researchers from both academia and industry have turned to developing efficient systems to detect fake media. Initial detection methods presented were mainly based on handcrafted features relating to inconsistencies resulting from the generation algorithm[37]. However, current methods primarily rely on deep learning techniques, such as CNNs or LSTMs, to identify manipulated footage. Below we provide an overview of the current datasets available alongside the existing approaches in this field, split into unimodal and multimodal, with a summary of all methods given in Table 3.2.

3.2.1 Existing Deepfake Datasets

In order to build AI methods capable of detecting deepfakes, there needs to be an appropriate dataset consisting of both real and fake footage that one can use to perform training and testing on. Summarised in Table 3.1, we provide an overview of the current popular datasets available for this task.

The **UADFV** dataset was one of the first available datasets, consisting of 49 real videos sourced from YouTube and 49 fake videos generated from FakeApp. The identity is always swapped to impersonate the actor Nicolas Cage.[52] The **Deepfake-TIMIT** dataset consists of 320 low-quality and 320 high-quality deepfake videos; all generated using faceswap-GAN based on the VidTimit dataset.[53] **FaceForensics++** offers 1,000 real videos, with 4,000 deepfake videos created using four different generative methods on the original videos (DeepFake, Face2Face, Faceswap and Neural Texture) [54, 55] **Celeb-DF** is a large scale dataset, offering 5,639 deepfake videos and 590 real ones. The videos are generated using an improved deepfake synthesis algorithm, which aims to reduce the visual artefacts that are present in other datasets.[56] Facebook's DeepFake Detection Challenge dataset (**DFDC**), first introduced as part of a Kaggle competition in early 2020, is the largest publicly available dataset, consisting of over 100,000 fake clips, generated using over 3,000 actors and a range of different generation algorithms. A total of 2,114 teams participated in the event, with the top scorer achieving 82.56% average precision[57]. It is crucial to note that this is the only dataset at present that contains fake faces, audio, or both. Lastly, the **Google/Jigsaw** deepfake detection dataset offers 3,068 deepfake videos generated from 363 original videos consisting of 28 actors, with the details of the deepfake generation not disclosed.[58]

Dataset	Real Videos	Fake Videos	Number of Actors	Real Video Source
UADFV	49	49	1	YouTube
DF-TIMIT	320	640	32	VidTIMIT Dataset
FaceForensics++	1,000	4,000	977	YouTube
CelebDF	590	5,639	59	YouTube
DFDC	23,654	104,500	3,426	Actors
DeepfakeDetection	363	3,068	28	Actors

Table 3.1: A comparison of the size of current publicly available deepfake datasets.

3.2.2 Unimodal Deepfake Detection Methods

Most existing deepfake literature focuses on one modality (visual input) to detect deepfakes. We now investigate unimodal methods to generate manipulated content, first using visual features and then audio.

Video Based

One of the earliest approaches to deepfake detection exploited that AI-generated content did not blink at the correct rate. Li et al. [59] highlight that the average person blinks at a rate of 17 blinks a minute, lasting for 0.1-0.4 seconds per blink. Deepfakes at the time, however, did not follow this pattern since most training sets did not contain faces with eyes closed. So by utilising a long-term recurrent convolutional network (LRCN) they can identify the irregular blinking that deepfakes hold. While conducting their experiments, no available datasets for deepfakes existed, so they created their own, which achieved an AUC score of 0.99 on their data. However, this approach is no longer sufficient since blinking has now been incorporated into deepfake generation.[60]

Demir and Çiftçi[61] earlier this year also approached deepfake detection by focusing on the eye region. They track geometric, visual, temporal, and spectral features of the eye to expose the eye gaze discrepancy deepfakes contain. They used the OpenFace library to extract their features and performed training on a network comprising of several dense layers. As a result, they achieved an accuracy of 92.5% on the FaceForensics++ dataset, 88.4% on CelebDF, and 99.3% on DeeperForensices.

How Do the Hearts of Deep Fakes Beat?[62] is the first paper to approach the problem using biological signals. They extracted Photoplethysmography (PPG) signals from the footage to identify the heartbeat and state that a person in a deepfake does not display a similar heartbeat pattern compared to a real video. This lack of consistent PPG signals in deepfake generation allows them to detect manipulated content. They achieved 93.4% accuracy on the FaceForensics++ dataset by using a VGG network (a type of deep CNN).

One forensic approach[63] focuses on identifying mismatches between phonemes and visemes. Phonemes are distinct units of sound, whilst a viseme is the visual counterpart, referring to the mouth shape to speak the phoneme. Agarwal et al. identify that the sound associated with the M, B and P phonemes require full mouth closure, which is not always present in deepfake content. On a custom dataset consisting of Barack Obama footage, they employ both a manual approach alongside a CNN model (Xception architecture) to achieve 97.0% accuracy on deepfake videos in the wild.

Despite the abundance of deep learning detection approaches, some have approached the problem using more straightforward supervised machine learning methods such as SVMs. For example, Yang et al.[64] highlight that where deepfake generation creates a face of a different person, whilst keeping the facial expressions of the actor, the two will have mismatched facial landmarks which can be exposed from their head position. They show that the head position estimated from their landmark positions is close for an original face, but will have a significant difference in the case of a deepfake video. They train an SVM on both the UADFV and DARPA datasets and achieve an AUC score of 0.89 and 0.84 respectively.

Similarly, Agarwal et al.[65] demonstrate that there exists a correlation between an individual's facial and head movements, which, since deepfakes expressions are being controlled by an impersonator, is disrupted for fake content. They extract facial action units and head pose in genuine footage for a specific person of interest and then train a one-class SVM, achieving an average AUC of 0.96.

Audio Based

Although we now look at how previous works have used audio features to detect fake media, we note that there have been no examples of using exclusively audio features on a dataset that contains audio-visual information. However, for completeness, we will explore how past works have detected generated audio.

Chen et al.[66] attempt to identify forged audio by utilising a large margin cosine loss function (LMCL) and a frequency masking augmentation to ensure their neural network architecture generalises to new spoofing algorithms. Furthermore, they extract low-level audio features (linear filter banks) on 30ms windows with a 10ms frameshift to achieve an EER of 1.26%.

Shan and Tsai[67] approach the problem of identifying tampered as well as spoofed audio, where tampered audio is the modification of original audio clips by the insertion, deletion, or replacement of content. They create a database of raw, unedited recordings of a person of interest collected from trusted and reliable sources. To classify a recording, they divide the audio into 25ms frames with 10ms frameshift. They compute the MFCCs, along with the Δ and $\Delta\Delta$ MFCCs for a total of 39 dimensions. Then they align the audio using the Needleman-Wunsch time-warping algorithm to match the clip with audio from the stored database. To classify each audio frame as matching or not, they employ an LSTM trained on recordings of Donald Trump (taken from the White Houses YouTube page) to achieve an EER of 0.43%.

In *A Comparison of Features for Synthetic Speech Detection*[68] the authors demonstrate that features representing spectral information, particularly in the high-frequency region, provide the best results when training a Gaussian mixture model. Lastly, Lieto et al.[69] employ CNNs to identify whether speech is generated from a bot or is a human. First, on a given audio signal, they pre-process it by applying peak normalisation, dividing it into frames with 50% overlap, and then apply a Hann window. Next, they compute the 2D image of a classical spectrogram plus the Mel-frequency spectrogram and provide these as inputs to the CNN. Overall, they achieve accuracy greater than 90% when trained on various datasets and demonstrate that the Mel-frequency spectrogram allowed for greater generalisation than the classical.

3.2.3 Multimodal Deepfake Detection Methods

By combining visual signals of media with the audio, we hope that they can provide complementary information and thus lead to stronger inferences, which we saw in Section 3.1.3.

Emotions Don't Lie[10] was one of the first approaches to adopt a multimodal method to deepfake detection. They extract perceived emotion from audio and visual modalities; they model the similarity (or disimilarity) between these modalities and their perceived emotion cues, where emotion belongs to the six discrete emotions commonly used in emotion recognition. Then, using a Siamese network-based architecture with their similarity-based metric, they provide a classification for the video. They train their network on both the DF-TIMIT and DFDC datasets since they are the only widely available datasets that contain audio in the clips. 2D facial landmarks, head pose, and eye gaze are the visual features chosen, extracted using the OpenFace library, whilst PyAudioAnalysis is used to obtain the first 13 MFCC coefficients. However, one requirement of their network is that training requires both a real video and its deepfake counterpart of the same subject to be passed. Overall, they achieved AUC scores of 96.3% / 94.9% on the LQ/HQ DF-TIMIT dataset, and 84.4% on the DFDC.

Chugh et al.[70] approached the detection problem by relying on dissonance, the lack of sync between audio and visual channels, and present their modality dissonance score, which is the metric used to label a video as real or fake. They used a deep network with the image crop for a frame as input for the visual ResNet stream and MFCCs as the audio features. They achieved results that improved upon *Emotion's Don't Lie*, achieving 97.9% / 96.8% AUC on the LQ/HQ DF-TIMIT dataset and 90.6% on the DFDC. One novelty this paper provides is temporal forgery localisation. Given footage, they can identify which frames in the video are real/fake, meaning that if only parts of the video were manipulated, their classification system could highlight these timestamps.

Gu et al.[71] also focus on the synchronisation between the audio-visual modalities. They partition the clips (which contain only real audio) based on the spoken phonemes and explore the visual defects in the mouth region to identify fake content. They use a CNN to capture the correlation of lip movements with the speech and then use a similarity metric to classify media. The visual input to their network is the extracted mouth regions, whilst the audio is Mel-scale spectrograms on a 40ms window. They achieved AUC scores of 99.2% and 97.4% on the LQ/HQ DF-TIMIT respectively.

Lastly, one of the top 25% of performing teams in Facebook's Kaggle competition using the DFDC dataset applied a multimodal approach to detection.[72] They combined information from images, video, audio, and the power spectrum of the image, training these on different models and fusing using a weighted average.

3.3 Summary

This chapter provided an overview of the literature in the affective computing domain, relating to emotion recognition through both facial expression and audio clips. We then reviewed the current advancements in deepfake detection, including the publicly available datasets, for both unimodal and multimodal approaches.

Paper	Detection Approach	Features	Classifier	Dataset	Score
In Ictu Oculi[59]	Unnatural Blinking Patterns	Eye Region	LRCN	Custom	0.99 AUC
Where Do Deep Fakes Look?[61]	Unnatural Eye Gaze	Eye Region and Gaze	MLP	FaceForensics++	92.5% Accuracy
How Do the Hearts of Deep Fakes Beat?[62]	Unnatural Heart Beat	PPG Signals	VGG	FaceForensics++	93.4% Accuracy
Detecting Deep-Fake Videos from Phoneme-Viseme Mismatches [63]	Phoneme-viseme Mismatches	Lip Region	Manual / Xception	Custom	97.0% Accuracy
Exposing Deep Fakes Using Inconsistent Head Poses[64]	Difference in Estimated Head Pose	Head Pose	SVM	UADFV DARPA	0.89 0.84 AUC
Protecting World Leaders[65]	Correlation Between Facial Expressions and Head Movements	Facial AU, Head Pose	SVM	Custom	0.96 AUC
Generalization Of Audio Deepfake Detection[66]	Creating Robust Feature Embeddings	Linear Filter Banks	ResNet	ASVspoof 2019	1.26% EER
A Cross-Verification Approach For Protecting World Leaders from Fake and Tampered Audio[67]	Needleman-Wunsch Alignment	MFCC, ΔMFCC, $\Delta\Delta$MFCC	LSTM	Custom	0.43% EER
Emotions Don't Lie [10]	Similarities Between Visual and Audio Modalities, and their Perceived Emotions	2D Facial Landmarks, Head Pose, Eye Gaze, and 13 MFCC	Siamese Network	DF-TIMIT DFDC	96.3% 84.4% AUC
Not Made for Each Other[70]	Audio-Visual Dissimilarity	Face Crop, 13 MFCC	ResNet, CNN	DF-TIMIT DFDC	97.9% 90.6% AUC
Deepfake Video Detection Using Audio-Visual Consistency[71]	Audio-Visual Dissimilarity	Mouth Region, Mel-spectrogram	CNN	DF-TIMIT	99.2% AUC

Table 3.2: A summary of current deepfake detection literature, providing their approach, features, and performance.

Chapter 4

Our Approach

This chapter explains the proposed multimodal models in detail, including the features selected, accompanied by the reasoning for these choices.

4.1 Overview

We tackle building a classifier to detect deepfake footage by extracting audio and visual elements that convey affective cues, which provide rich emotional information, and then pass these to a hybrid deep neural network. Given an input video, we aim to classify it as real or fake. Moreover, we investigate the use of different feature fusion methods (early, mid-layer, or late) on our hybrid networks consisting of CNNs, LSTMs or MLPs.

4.2 Features

As previously introduced, we offer a multimodal approach to the problem as it has been shown that the use of more than one modality can provide complementary information, thus generating richer models. For example, Gunes and Piccardi in [73] observe that when modalities are projected into a shared space, they point to similar affective cues. Therefore, we select relevant audio-visual features based on emotion recognition and affective computing research. Whilst our approach is similar to *Emotions Don't Lie*[10], as they also discuss affective computing in their paper, we do not classify our audio-visual streams into discrete emotions. Instead, we believe the classifier should infer the relationship between these affective cues without the explicit mapping to discrete emotional space, which allows for richer information since emotions are not discrete in real life. Whatsmore, manipulated content need not be emotional; if the content is generated and the actor neutral, then their approach may fail whilst ours would still hopefully be capable of detecting the deepfake.

We will now turn to discuss the visual and audio features we have selected, inspired by the literature reviewed in Section 3.1. An overview of these chosen features can be found in Table 4.1.

4.2.1 Visual Features

We stated previously that the facial action coding system (FACs) was the prefered way to encode facial features in affective computing. Therefore, we incorporate this system by using the facial action units (AUs) to track the muscle movement of a subject. These AUs have been used extensively in facial expression recognition and were introduced to aid deepfake detection by Agarwal et al.[65]

We also incorporate eye gaze and head position into the model. Firstly because numerous of the deepfake papers we reviewed in Section 3.2.2 included either one of these elements, which highlights their importance, as these inputs often contain inconsistencies in fake videos as a result of their generation. In addition, although eye gaze and head position have not been as well explored as facial landmarks for the task of expression recognition, they have been shown to provide complementary information. For example, [47] examines their use in addition to facial appearance to, at the time, outperform the state-of-the-art in facial expression recognition, demonstrating the value that these extra cues can offer.

We use the open-source facial behaviour analysis toolkit OpenFace2[74] to extract our desired visual features. In total we obtain 31 visual characteristics, consisting of:

- Intensity of the 17 AUs the library can extract, rated on a scale from absent to maximal of non-overlapping facial muscle actions
- 8 gaze related features including the angle and direction for both eyes
- 6 head-pose features (location and rotation)

4.2.2 Audio Features

Generally, raw speech data includes many spikes and background noise that are not good inputs to a learning model. Thus, we need to extract the data provided in the utterance into a suitable input form. In addition, audio files should undergo various pre-processing steps, such as applying a window (which was covered in Section 2.1) to ensure we avoid degradation of the signal.

We have decided to use MFCC, a short-term spectral feature, as our audio input. MFCCs are used frequently in works investigating audio or speech signals for perceived emotion recognition or speaker identification. Whatsmore, MFCCs have been considered state of the art for analysing speech signals for over three decades.[70]

Feature	Description	Feature	Description
gaze_0_x	Eye gaze direction vector for eye 0 on x axis	AU05_r	Intensity of Upper Lid Raiser
gaze_0_y	Eye gaze direction vector for eye 0 on y axis	AU06_r	Intensity of Cheek Raiser
gaze_0_z	Eye gaze direction vector for eye 0 on z axis	AU07_r	Intensity of Lid Tightener
gaze_1_x	Eye gaze direction vector for eye 1 on x axis	AU09_r	Intensity of Nose Wrinkler
gaze_1_y	Eye gaze direction vector for eye 1 on y axis	AU10_r	Intensity of Upper Lip Raiser
gaze_1_z	Eye gaze direction vector for eye 1 on z axis	AU12_r	Intensity of Lip Corner Puller
gaze_angle_x	Eye gaze direction in radians (left-right)	AU14_r	Intensity of Dimpler
gaze_angle_y	Eye gaze direction in radians (up-down)	AU15_r	Intensity of Lip Corner Depressor
pose_Tx	Location of the head with respect to camera in millimeters (x axis)	AU17_r	Intensity of Chin Raiser
pose_Ty	Location of the head with respect to camera in millimeters (y axis)	AU20_r	Intensity of Lip Stretcher
pose_Tz	Location of the head with respect to camera in millimeters (z axis)	AU23_r	Intensity of Lip Tightener
pose_Rx	Head position rotation in radians on x axis (pitch)	AU25_r	Intensity of Lips Part
pose_Ry	Head position rotation in radians on y axis (yaw)	AU26_r	Intensity of Jaw Drop
pose_Rz	Head position rotation in radians on z axis (roll)	AU45_r	Intensity of Blink
AU01_r	Intensity of Inner Brow Raiser	MFCC 0-12	First 13 Mel-Frequency Cepstral Coefficients
AU02_r	Intensity of Outer Brow Raiser	ΔMFCC 0-12	Differential of MFCC
AU04_r	Intensity of Brow Lowerer	ΔΔMFCC 0-12	Acceleration of MFCC

Table 4.1: Names and their respective description of the 70 features we have chosen.

Taking the first 13 cepstral coefficients, along with the first 13 for ΔMFCC and ΔΔMFCC, we obtain a total of 39 audio features. We take the derivatives of the MFCC since they provide us with the dynamics of the power spectrum and have been shown to offer substantial improvements to the task of speaker recognition than if just the MFCC was used.

4.3 Classification Architecture

Once we had decided on the chosen features, we next consider how to combine these to implement the classifier. One of the difficulties with using multiple modalities is knowing how and when to combine them. Therefore, we now look at each potential option to fuse the modalities, including a unimodal approach, acting as a baseline. Finally, we plan to implement each method to evaluate which fusion mode is the best approach. Figure 4.1 provides a diagram of our chosen architectures.

4.3.1 Unimodal Learning

We begin our experiments with a unimodal approach, testing audio and visual separately, to achieve a baseline that we can hope to improve when implementing our multimodal networks.

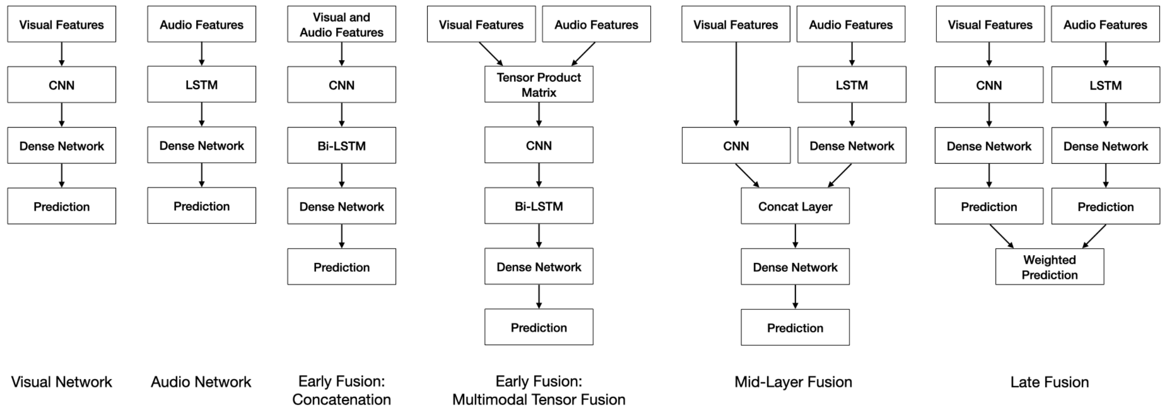

Figure 4.1: Diagrams of our chosen classification architecture for our unimodal and multimodal approachs, and each method of feature fusion to be investigated.

Visual

Using only the given 31 visual features, we use a series of CNNs and dense layers to classify a video based purely on the visual modality. CNNs were the most appropriate architecture given their extensive use for visual approaches in the previously reviewed affective computing and deepfake literature.

Audio

For our 39 audio features, we implement an LSTM followed by a series of dense layers to provide our audio prediction. We chose an LSTM to classify the audio stream since it has been heavily used for speaker emotion recognition.

4.3.2 Early Fusion

Early fusion (or feature-level fusion) combines the modalities into a single input feature vector. We explore two early fusion approaches, one additive and one multiplicative.

Concatenation

We begin our multimodal feature experiment by simply concatenating visual and audio features as input to achieve a 70-dimensional feature vector. We note that additive combinations such as this concatenation assume that every modality is always potentially useful, which means this model may be sensitive to sensor noise.[12]

We pass this input through a series of CNNs to extract the abstract representations of the features, followed by a BiLSTM to allow for sequence learning across the whole 10-second video clip. Lastly, there are a series of dense layers to produce the output.

Multimodal Fusion Tensor Network

Prior works have explored using multiplicative combination methods[75], which can explicitly represent the relative reliability of each modality. Using the approach given by *Zadeh et al.[76]*, we apply the following formula to our vectors, referred to as a multimodal tensor fusion network.

$$h_m = \begin{bmatrix} h_x \\ 1 \end{bmatrix} \otimes \begin{bmatrix} h_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_x & h_x \otimes h_y \\ 1 & h_y \end{bmatrix}$$

Where h_x is the visual features vector for a frame, and h_y the audio, resulting in a (32,40)-dimensional matrix. Thus, performing this operation over n frames in a given file results in the input being a 4D matrix. This matrix is then provided as input to a network with the same structure as the additive approach to early fusion.

4.3.3 Mid-layer Fusion

Mid-layer fusion takes several separate vector inputs, creates deep representations of each, and then combines these using a concatenation layer to achieve a multimodal vector which acts as the input to a series of more layers. In our case, we have two inputs, the audio and visual streams, trained on different networks and then combined and processed through several more layers.

We perform mid-layer fusion by inputting the visual features through a CNN, and the audio to an LSTM, then concatenating these deep representations and passing this through a dense network to produce the output prediction.

4.3.4 Late fusion

Late fusion, sometimes referred to as decision level fusion, consists of a series of separate unimodal classifiers trained on each modality. Each classifier forms a prediction based on their modalities input stream, and these predictions are then combined to achieve the final prediction. Late fusion, therefore, learns different representations for each modality; however, one disadvantage is that it cannot learn cross-modal correlations.

We first produce predictions for the visual and audio stream individually using their unimodal classification networks to perform our late fusion. Then, we combine the two arrays to provide the final predictions. Some methods use simple averaging by taking the mean of the two classifiers to achieve the final score. Here, we use a weighted classifier, which calculates the score in the following way:

$$\text{Weighted Score} = \alpha \times \text{Visual} + (1 - \alpha) \times \text{Audio}$$

Where visual is the output score from the unimodal visual model, and audio from the unimodal audio model. Note that if $\alpha = 0.5$ this is the same as taking the mean. If $\alpha = 0$ this is the unimodal audio case, and similarly if $\alpha = 1$ the completely visual. We will find the optimal value of α by testing different values and selecting that with the top AUC score.

4.4 Summary

This chapter has presented the 31 visual features and 39 audio features that we use for deepfake detection. Furthermore, we covered the networks we will implement to perform detection, consisting of different types of feature fusion (early, mid-layer, or late).

Chapter 5

Implementation

This chapter presents the project's technical details, including preparing the dataset for training, extracting the desired features, and implementing the chosen deep learning algorithms.

5.1 Dataset

The DFDC dataset contains over 100,000 10 second video clips from 3,426 different actors. This extensive dataset results in the training split of provided videos containing over 450GB of raw footage. An example of a genuine frame and its manipulated counterpart from the dataset can be seen in Figure 5.1.

Due to the size of the dataset, we have conducted testing on a small subset of the training set, decreasing compute time, allowing us to receive results faster. Moreover, only a fraction of the training dataset contains falsified audio, and so we wish to ensure we include these clips. The dataset is partitioned into 50 folders, all almost equal in size, with the clips that potentially contain manipulated audio in the last five folders. Therefore, we include the last five folders as the dataset. Whatsmore, when loading in the data, we discovered that the real to fake value counts were heavily skewed towards fake footage (only 14% of data was labelled real). Thus, we included additional real labelled footage from other

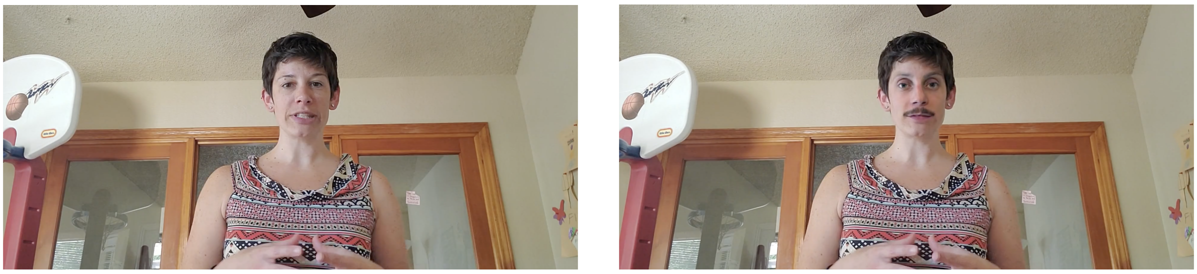

Figure 5.1: Example of real (left) and fake (right) frames from the DFDC dataset.

folders in the dataset to make the value counts more even, avoiding an imbalanced dataset. As a result, we load in a total of around 21,000 videos, around 18% of the full dataset.

5.2 Feature Extraction

All the videos in the DFDC dataset are of .mp4 format and are 10 seconds long. Since it is filmed in 30 frames per second, each video clip should result in 300 data points. We begin by using the OpenFace 2.0 library[74], an open-source facial behaviour analysis toolkit to extract the visual features in each video. Figure 5.2 demonstrates the tracking of facial landmarks, head position and eye gaze from this library. The library has also been trained to extract the intensity of a selection of action units for a frame. In total, we have 31 visual data points for each of the 300 frames in a clip.

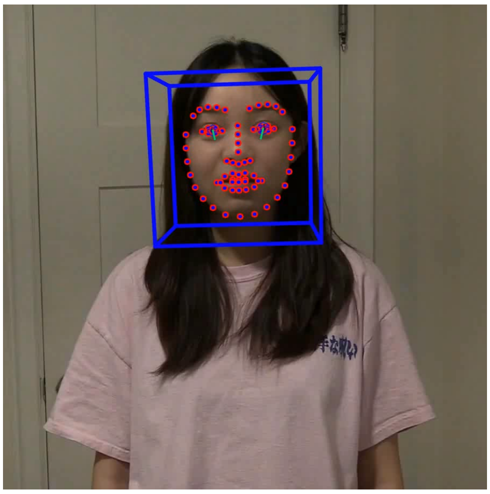

Figure 5.2: An example of OpenFace 2.0 libraries facial landmark, gaze, and head pose tracking.

Simultaneously, for each frame, we also must extract the audio features. We begin applying pre-emphasis to the signal, sampled at 44.1kHz; pre-emphasis is calculated as

$$y(t) = x(t) - \alpha x(t-1)$$

where $\alpha = 0.97$. Next, we split the audio file into windows, each 2048 samples (46ms) long, with a frameshift of 1467 samples (33ms) to match the video footage. This window size provides each audio frame with a 561 sample (13ms) overlap, sufficient in size to reduce artifacts during FFT calculations. Ideally, we would have prefered to have a more typical window size such as 25ms duration with 10ms frameshift. However, we required the same number of audio samples as video, which needed to match the videos sampled at 30 frames per second.

Then, to reduce the chance of spectral leakage, we must apply a windowing function to the frames. Here we use the most commonly applied window, the Hamming window, given as:

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$

Where $0 \leq n \leq N-1$ if N is the window length (2048 samples).

Finally, we use the `python_speech_features`[1] library to extract the first 13 Mel-Frequency

[1] `https://python-speech-features.readthedocs.io/en/latest/`

Cepstral Coefficients (MFCCs) and their respective ΔMFCCs, and $\Delta\Delta$MFCCs. These steps provide us with a 39-dimensional audio feature vector for each frame in a given file.

We have now obtained our 70 input features for 300 frames of each file. We ran this feature extraction process separately for each folder in the training dataset, saving the results as a .csv file. Alongside the features, we also included the filename and frame number to identify which video each row belongs to.

5.3 Data Preperation

We then load in the .csv files for our chosen subset of folders, and as previously mentioned, we also load in more files with the real label from the remaining folders to counter the skew in proportions. These extra files increase the percentage of real values in the dataset from 14% to 48%. Next, we then explored the values in the dataset, removing any rows that contain null, infinite, or not a number (NaN) values. Next, we split the dataset into 80:10:10 of train/validate/test, providing us with 17,000 files to perform testing on, and 2,100 for both validation and testing.

The last step in pre-processing is normalisation. Normalising the features brings the values into the same range to ensure they contribute an equal amount to the classification, since some values are between [0,∞], whilst others [0,1]. Using the python `scikit-learn` library, we performed **min-max normalisation** which rescales all the features into the [0,1] range using the following formula:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Since we wish the input to the classifier to be a sequence of frames rather than a collection of total frames, we must reshape the data frame. We, therefore, group arrays by their filename and create an output matrix with the following dimensions (total number of files, 300, 70), allowing us to perform training on a whole file rather than individual frames.

5.4 Classification Network Architecture

We use the Keras library[2], a popular Python deep learning library built on top of TensorFlow to implement our networks. All models are trained using the Adam optimiser, with a batch size of 64 and early stopping on the validation loss with a patience of 15. We use a sigmoid activation function with binary cross-entropy loss in the output layer to provide the probability that the video is real or fake.

[2] `https://keras.io`

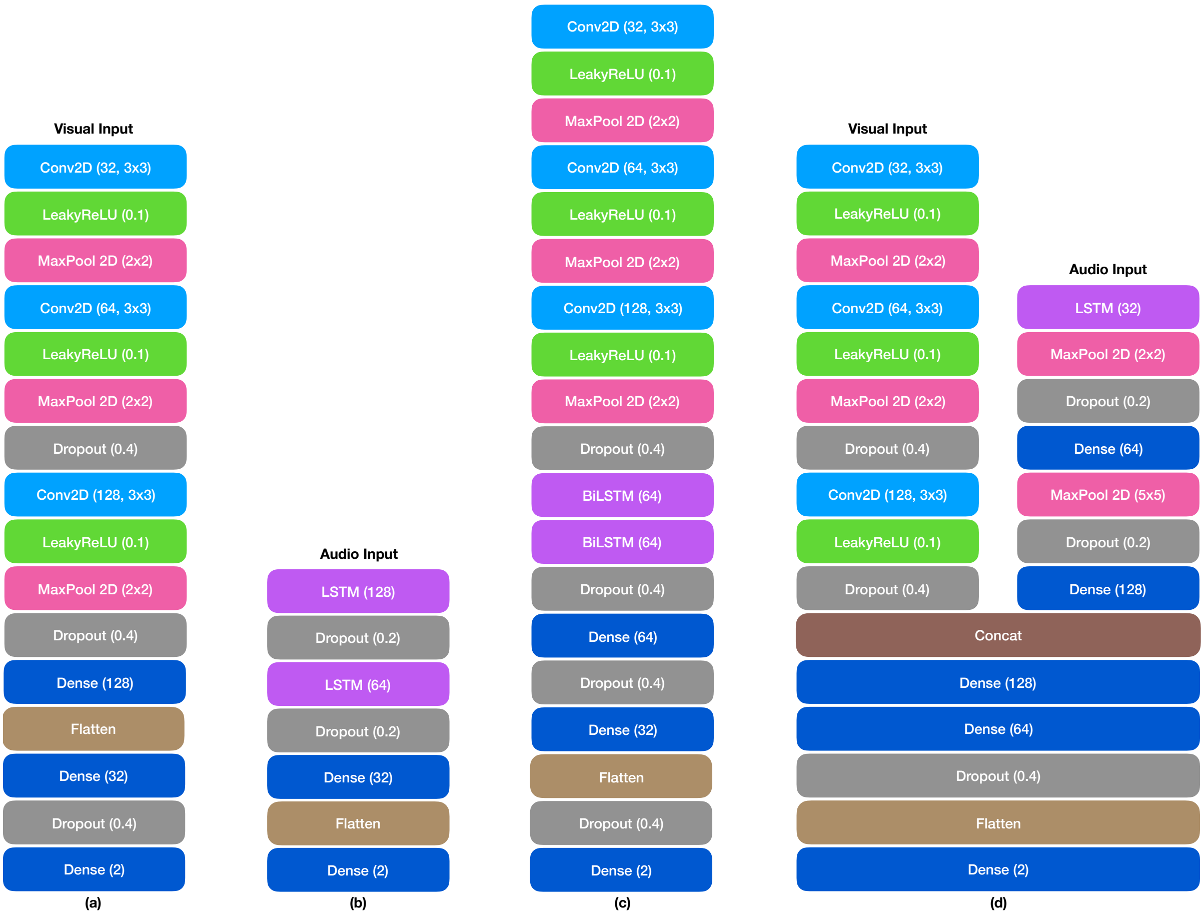

Figure 5.3: Architecture diagrams of our final networks. (a): unimodal visual (b): unimodal audio (c) early fusion, input either additive or multiplicative of audio visual features (d): mid-layer fusion

Figure 5.3 displays a diagram of the four different deep learning networks we implemented for the different models we previously introduced in Figure 4.1. Here, (c) is the early fusion approach, and the input can either be the concatenation of the modalities, a 70-dimensional vector, or the multimodal tensor fusion network, a 32 × 40 input. The final fusion approach we test, late fusion, uses a weighted classifier on the output from the visual and audio models - hence no diagram. Instead, below we give the function used to generate the predicted output for the late fusion:

```
def get_weighted_prediction(vis_predicted_y, audio_predicted_y, alpha):
    return alpha*vis_predicted_y + (1-alpha)*audio_predicted_y
```

5.5 Summary

This chapter has discussed the dataset we selected and the pre-processing steps needed to transform the videos into our desired input form. We also covered the technical details of our deep learning network.

Chapter 6

Results and Discussion

In this chapter, we discuss the results of our implementation, comparing the six different classification architectures on our validation dataset. Then, we decide on our final method from these results and test it on our hold-out data. We also test our model on another dataset before comparing it to the state of the art. Lastly, we investigate which features were the most important in classification.

6.1 Initial Testing

Using the evaluation metrics discussed in Section 2.2.2, we compare the proposed different approaches, both unimodal and multimodal, and the four different modality fusion techniques. Further results from these experiments can be found in Appendix A.

As shown in Table 6.1, the additive approach to early fusion, simply concatenating the audio-visual features, offered the best accuracy and AUC. In addition, Figure 6.1 yet again shows that this early fusion via concatenation was the best performing approach as its ROC plot (coloured green) has the best response curve. Moreover, we believe this fusion method performed the best since, in training, the network offers a shared representation for the audio-visual features since they are integrated from the start, making it consistently multimodal.

We observe that not all fusion techniques improved upon the unimodal approaches intended to act as our baseline. The multimodal fusion tensor network (MFTN) was the

Method	Unimodal (Visual)	Unimodal (Audio)	Early Fusion (Concat)	Early Fusion (MTFN)	Mid-Layer Fusion	Late Fusion
Accuracy	84.4	83.2	**89.7**	80.9	86.7	89.2
AUC	91.0	91.3	**96.2**	89.6	94.0	94.2

Table 6.1: Accuracy and AUC scores of different methods implemented on the validation data.

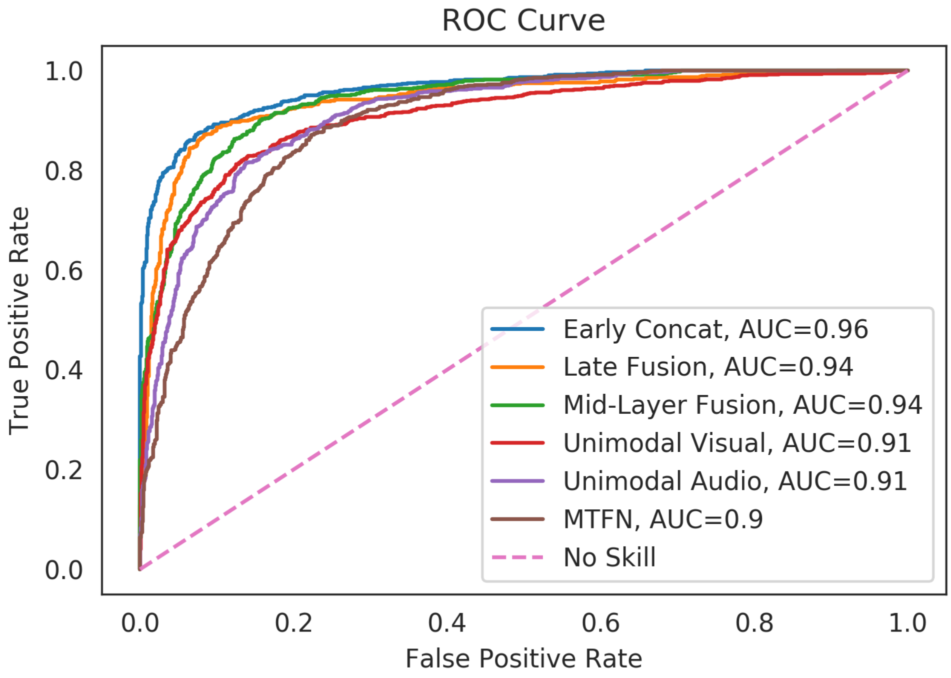

Figure 6.1: A plot of ROC curves for the six different methods tried, tested on the validation data.

worst-performing model applied. We now look at the correlations between the audio-visual features before we can reason about the performance of these different fusion methods.

6.1.1 Correlation of Features

Figure 6.2 shows a heat map of the correlation of our features, calculated using the Pearson correlation coefficient. It is immediately apparent that distinct clusters exist in this graph, areas demonstrating high (positive or negative) correlation, and these regions exist mainly between the same modality. To aid explanation, areas that contain an absolute correlation score of more than 0.3 have been labelled, and below follows a description of how to interpret each area.

Area 1: Facial AUs

Firstly, we see that the facial action units are correlated amongst themselves. This is to be expected since the movement of one muscle group is likely to cooccur with others. For example, AU6 (cheek raiser) is positively correlated with AU12 (lip corner puller) since an expression such as smiling would raise the intensity of both these muscle groups.

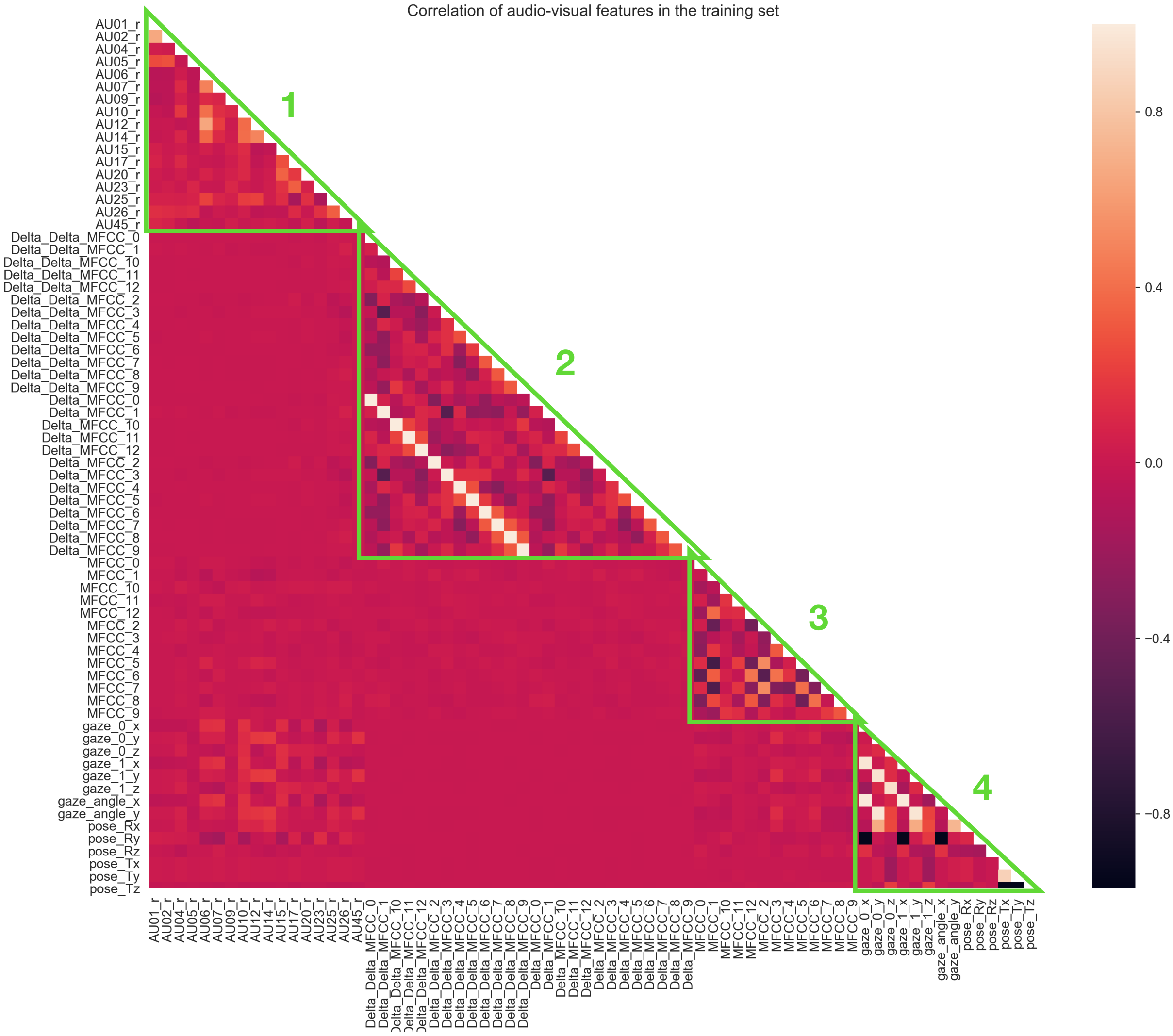

Figure 6.2: A Heat Map to show the Pearson Correlation of features in our training dataset. Areas of high correlation (> |0.3|) have been outlined in green, correlation pairs are: 1: AUs, 2: Delta and delta-delta MFCCs, 3: MFCCs, 4: Pose and gaze.

Area 2 and 3: MFCCs, Delta MFCCs, Delta-Delta MFCCs

The large triangle in the middle region (area 2) contains the correlation between the Δ and $\Delta\Delta$MFCC features. The white diagonal line represents the perfect correlation between the ΔMFCC and its respective $\Delta\Delta$, which is expected due to the differential calculation. Similarly, in area 3, the MFCC correlations exhibit strong correlations amongst themselves. Unable to explain why there are strong negative correlations amongst these MFCC features, we further divide the audio correlation matrix into real and fake footage, as pictured in Figure 6.3. Analysing real and manipulated content independently exposes that the strong negative correlations amongst the MFCC features in deepfake footage but

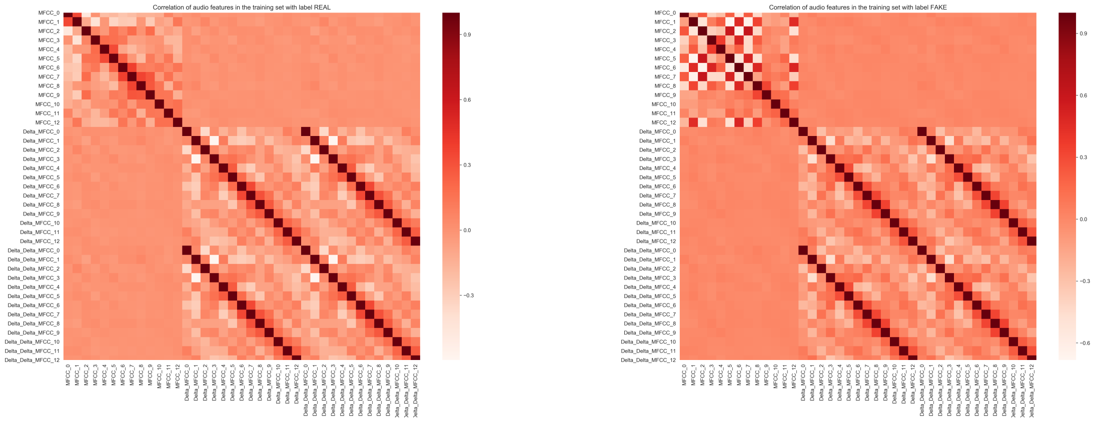

Figure 6.3: Heat map of Pearson Correlation on audio features in the training dataset with labels real (left) and fake (right).

not real. These must result as artifacts of audio manipulation, which is why they are not present for real audio samples. Overall, we emphasise that the correlations of audio features only exist within the audio modality - there are no cross audio-visual correlations.

Area 4: Gaze and Head Pose

The last area in Figure 6.2 we discuss is the correlation amongst the gaze and pose features. There is a positive correlation amongst the gaze features; we expect the two eyes to work together, resulting in the eye-tracking for both the left and right eye to be coordinated. Moreover, the correlation between eye gaze and head pose has likely arisen as a side effect of recording the footage. Although the dataset does contain people facing away from the camera, many videos have the actor facing towards the camera, meaning their head and eyes would face the same way.

To conclude this diversion from our results, we have shown that there exists no correlation between the audio-visual features, but instead, all correlations occur within the given modality. Furthermore, the correlations also occur within the same type of feature in a modality, such as the facial AUs, MFCCs, and gaze. Additionally, we have provided reasoning for the occurrence of these correlations.

6.1.2 Inital Results Explination

We now return to discuss our initial results and explain why the multimodal fusion tensor network for early fusion had the worst performance across all methods. When we perform early fusion using the MFTN approach, to combine the modalities, we take the outer product of the visual vector with the audio, given as $h_x \otimes h_y$ where h_x is the visual

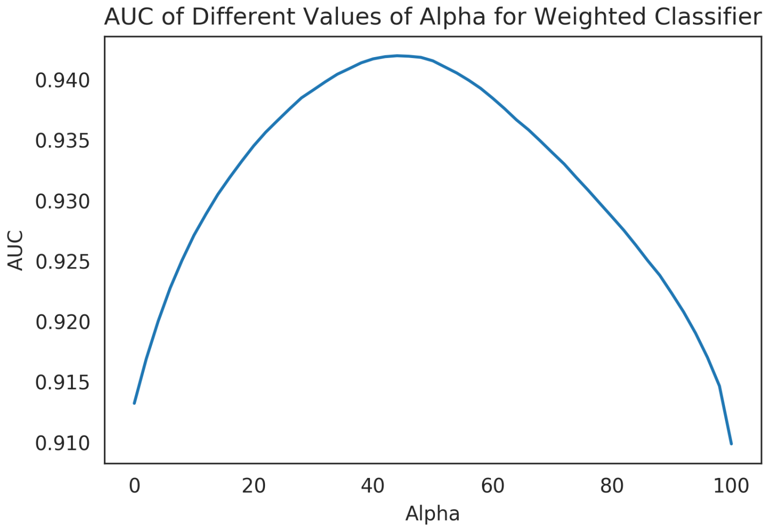

Figure 6.4: A graph to show the effect alpha has on the AUC score of our weighted classifier when tested on the validation data. Alpha is optimal at 44%.

features and h_y the audio, resulting in a large matrix of their products. However, since we just stated that there are no strong correlations between the two modalities, performing $h_x \otimes h_y$ increases the dimensionality from 70 input features per frame to 32×40, but does not provide beneficial input information since there are no audio-visual correlations. Therefore, this high dimensionality of data ends up being a worse input than either modality individually.

Additionally, the absence of correlation between the audio and visual features is why the weighted classifier performed highly, achieving the second-best scores. The reasoning for this is that each modality will learn a separate representation, and since there is very little correlation between modality, minimal information is lost. Thus, when we perform classification, we combine the individual strengths of each modality and achieve nearly as high an accuracy as our best method. To perform late fusion, as previously introduced we used a weighted classifier to achieve a final score, using the following formula:

$$\text{Weighted Score} = \alpha \times \text{Visual} + (1 - \alpha) \times \text{Audio}$$

Where visual and audio are the outcome of the respective unimodal classifiers, a number between [0,1] where a score of zero means the model believes it is completely real and one completely fake. Therefore, the weighted score is a combination of these two scores where if $\alpha = 0$ we have the purely audio score, or if $\alpha = 1$ the visual. Figure 6.4 shows how we found the optimal α value, which is given at 0.44.

To summarise, our additive early fusion approach achieved the best scores and will be chosen as our final model. The weighted classifier was the second highest achieving model,

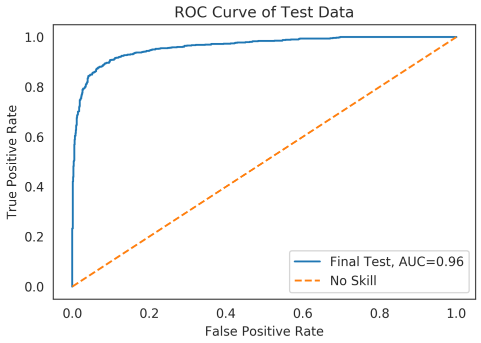

Figure 6.5: A ROC curve of the final models performance on the hold-out test data

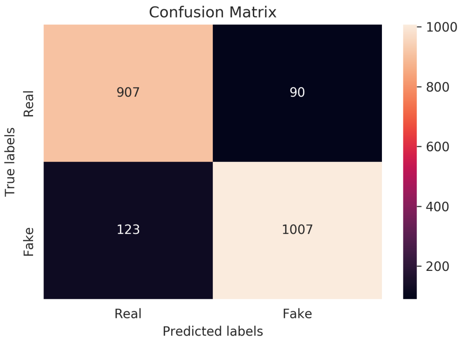

Figure 6.6: The confusion matrix of the results from our final models classification on the hold-out test data

and this shows that although the modalities lack correlation, there must be a high level shared audio-visual representation that the additive early fusion learns to outperform the late fusion approach. The similarity in scores between the unimodal approaches highlights that both audio and visual features are important in classifying deepfakes. Finally, the multiplicative fusion approach's performance reminds us that they should be considered a one-dimensional vector since the two modalities lack correlation.

6.2 Evaluation of Final Model

We now turn to test our chosen model, a CNN-BiLSTM network with a 70 audio-visual feature vector as input, on our hold-out test data.

	Precision	Recall	Accuracy	AUC
Validation Data	**89.6**	89.0	89.7	**96.2**
Test Data	89.1	**90.7**	**90.4**	**96.2**

Table 6.2: Precision, Recall, Accuracy, and AUC scores of our validation data and hold-out test data on our final chosen model.

Table 6.2 provides an overview of the evaluation metrics for both the validation data and our hold out data. Given that there is no significant difference between these figures across the two datasets, we can be confident that our model has not overfitted to either the training or validation set, thus generalising to new unseen data.

Figure 6.5 shows the ROC curve for our models predictions on the test dataset, while Figure 6.6 provides the confusion matrix for these predictions. From this, we can see that there are slightly more occurrences of manipulated content being predicted as real rather than genuine footage being predicted fake. These misclassifications of fake footage likely

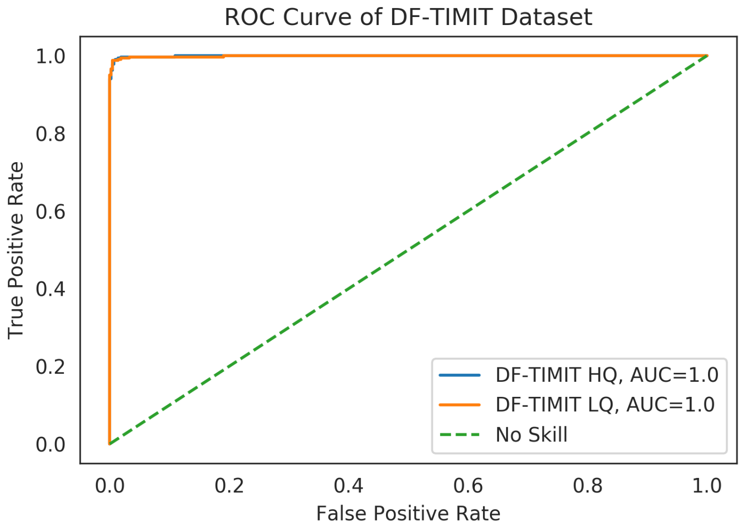

Figure 6.7: A ROC curve of the final models performance on the DF-TIMIT dataset.

occur since this dataset deliberately contains obstructions, different lighting conditions, and actors not facing the camera. All of these makes it harder for us to extract our chosen visual features, and in turn, the model classifies the video as real.

6.3 Testing on DF-TIMIT Dataset

Next, to assess the generalisation of our model, we test it on other available data. The only other publicly available dataset reviewed in Table 3.1 that has audio in the clips is the Deepfake TIMIT dataset[53], therefore this is the only other dataset suitable for us to test both modalities on.

The DF-TIMIT dataset includes 640 manipulated videos, containing two versions of the same video, one low quality and one high quality. Hence, the dataset is split into DF-TIMIT LQ, consisting of 320 low-quality fake videos, and DF-TIMIT HQ, 320 high-quality versions of the same fake video. Although the videos contain audio, no manipulation has been done to the audio channel, unlike some of the videos in the DFDC dataset. The real footage (and what is used to generate the deepfakes) is from the Vid-TIMIT database, a collection of videos created to aid research on topics such as automatic lip reading and multimodal speech recognition.[77] It contains 43 actors, each reciting 10 pre-selected short sentences.

Figure 6.7 displays our model's performance when tested on both the DF-TIMIT HQ and DF-TIMIT LQ. We were pleased to see that we achieved a near-perfect AUC score of 99.9 on both versions. We also achieved 90.4 / 90.5 % accuracy on the LQ and HQ versions respectively, matching what we achieved on our original test data.

We previously stated that this dataset consists of no manipulated audio, however, our

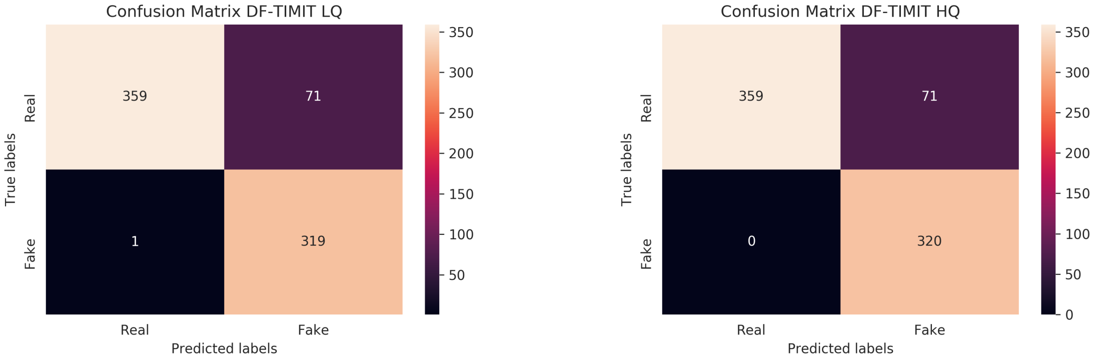

Figure 6.8: The confusion matrices from testing our model on DF-TIMIT LQ (left) and DF-TIMIT HQ (right).

performance on this dataset suggests that perhaps the audio features are capable of detecting more than manipulated audio, but possibly also aid the learning of audio-visual inconsistencies, similar to [71] discussed in Section 3.2.3, who use the correlation of lip movements with speech on the DF-TIMIT dataset to achieve 99.2% and 97.4% on the LQ/HQ respectively. Overall, these results instil confidence in our model's ability to detect deepfakes in the wild.

We see from the confusion matrices in Figure 6.8 that the model perfectly identifies all instances of deepfake videos in the HQ set, and all but one in the LQ. Thus, we theorise that the DF-TIMIT dataset is 'easier' to detect fake footage than the DFDC dataset. Firstly, the original footage (VidTIMIT database) was shot in a controlled environment, where each actor stands in front of the same backdrop with the same lighting conditions and faces the camera to speak. On the other hand, the DFDC dataset consists of videos in an uncontrolled environment, both indoor and outdoor settings, in various lighting conditions. This lack of consistency in the DFDC makes it inevitably harder to identify videos since some scenes may be too dark or have their head turned fully sideways for the library to extract our facial features correctly.

Next, the DF-TIMIT dataset only uses two different generation algorithms to create their deepfakes, whilst the DFDC uses four times as many. Furthermore, the facial region output resolution when generating the deepfakes is higher for the DFDC (256x256, vs 128x128 for HQ and 64x64 for LQ DF-TIMIT). The higher quality the face crop is, in theory, the better the deepfake. Lastly, the DFDC set deliberately contains obstructions, noise, filters and other augmentations whereas the DF-TIMIT does not. This means it is harder to identify footage in the DFDC due to these distractions.

Consequently, we state that DFDC is objectively a better dataset than DF-TIMIT. However, in fairness, DF-TIMIT was created at the end of 2018, and the DFDC in 2020. Hence, much advancement in this area had taken place in the two-year difference between their release, so one expects the latter to be of better quality.

Method	Modality	Datasets		
		DF-TIMIT		DFDC
		LQ	HQ	
Capsule-forensics[78]	V	78.4	74.4	53.3
Multi-task Learning[79]	V	62.2	55.3	53.6
Inconsistent Head Pose[80]	V	55.1	53.2	55.9
Two-stream Neural Networks[81]	V	83.5	73.5	61.4
Exploiting Visual Artifacts[82]	V	77.0	77.3	66.2
MesoNet[83]	V	87.8	68.4	75.3
Xception-c23[84]	V	95.9	94.4	72.2
Face Warping Artifacts[85]	V	**99.9**	93.2	72.7
Texture Features[86]	V	92.6	94.4	79.5
Emotions Don't Lie[10]	AV	96.3	94.9	84.4
Not Made for Each Other[70]	AV	97.9	96.8	90.6
Our Method	AV	**99.9**	**99.9**	**96.2**

Table 6.3: Comparison of AUC scores of our method with other techniques on both the DFDC and DF-TIMIT datasets.

As a result, given that we trained our model on the DFDC, it creates a network capable of generalising to new environments and noisy conditions better than those trained on the DF-TIMIT. Thus, when we test it on DF-TIMIT, this data is more straightforward than what it was trained on, resulting in a near-perfect AUC score.

6.4 Comparison with SOTA Methods

Table 6.3 compares the performance of our method with the state-of-the-art. Our method is the top across all three datasets and achieves over 5% higher AUC on the DFDC dataset. We also observe that generally, the audio-visual approaches outperform the purely visual ones, highlighting that multimodal approaches are superior since the audio and visual streams can provide complementary information and therefore create more robust inferences about the media. Furthermore, these results also highlight our earlier point that the DF-TIMIT is an 'easier' dataset since the AUC score for DFDC is often considerably lower. Hence, we propose that future works be trained on DFDC rather than DF-TIMIT to ensure a model that can generalise better to videos in the wild.

We previously stated that our approach to deepfake detection was inspired by *Emotion's*

Feature	Description	Feature	Description
`gaze_0_x`	Eye gaze direction vector for eye 0 on x axis	`AU07_r`	Intensity of Lid Tightener
`gaze_0_y`	Eye gaze direction vector for eye 0 on y axis	`AU10_r`	Intensity of Upper Lip Raiser
`gaze_1_x`	Eye gaze direction vector for eye 1 on x axis	`AU12_r`	Intensity of Lip Corner Puller
`gaze_1_y`	Eye gaze direction vector for eye 1 on z axis	`AU14_r`	Intensity of Dimpler
`gaze_1_z`	Eye gaze direction vector for eye 1 on z axis	`AU15_r`	Intensity of Lip Corner Depressor
`pose_Rx`	Head position rotation in radians on x axis (pitch)	`AU17_r`	Intensity of Chin Raiser
`pose_Ry`	Head position rotation in radians on y axis (yaw)	`AU20_r`	Intensity of Lip Stretcher
`AU01_r`	Intensity of Inner Brow Raiser	`AU26_r`	Intensity of Jaw Drop
`AU04_r`	Intensity of Brow Lowerer	`MFCC 0-8`	First 9 Mel-Frequency Cepstral Coefficients
`AU06_r`	Intensity of Cheek Raiser	`MFCC 10-12`	10-12 Mel-Frequency Cepstral Coefficients

Table 6.4: The 'best' 30 features in our training data, selected based on chi-squared test scores.

Don't Lie[10], so we are satisfied that we significantly improved on their scores. This performance increase arises from not mapping our audio-visual features to six emotions and instead relying on extracting features known to convey emotional cues and leaving them in their original form. Furthermore, our method is easier to extend to new training data since their model required both a real and fake video of an individual at training time, whereas we have no such requirement.

6.5 Top Features

To aid future work, we perform a final investigation into which features out of our original 70 are of most importance; we wish to discover which are the top features providing most of the information to classify a video as real or fake.

We apply the **Chi-Squared test** to the training data, which tests the independence of a feature variable with the class label. If the feature variable and class label are independent, then the feature variable is not relevant in deciding the class label, and so we can discard it. The `SelectKBest` function from the `scikit-learn` library allows us to apply the chi-squared algorithm, and then select the top 'K' most dependent features.

Figure 6.9 shows the cumulative scores for the sorted feature scores produced by the chi-squared algorithm. The red line represents 99% of the cumulative feature scores, meaning that the features which lie to the right of the intersection with that line are adding very little information, and so we can remove them for a reduced dimensionality, hopefully without affecting the performance significantly. We run the `SelectKBest` algorithm, where K=30 to achieve 30 new features (given in Table 6.4) for which we experiment with, consisting of 18 visual and 12 audio.

We train these new features on the same architecture and training data as before, just adjusted to now input 30 features instead of 70. The performance of this reduced feature

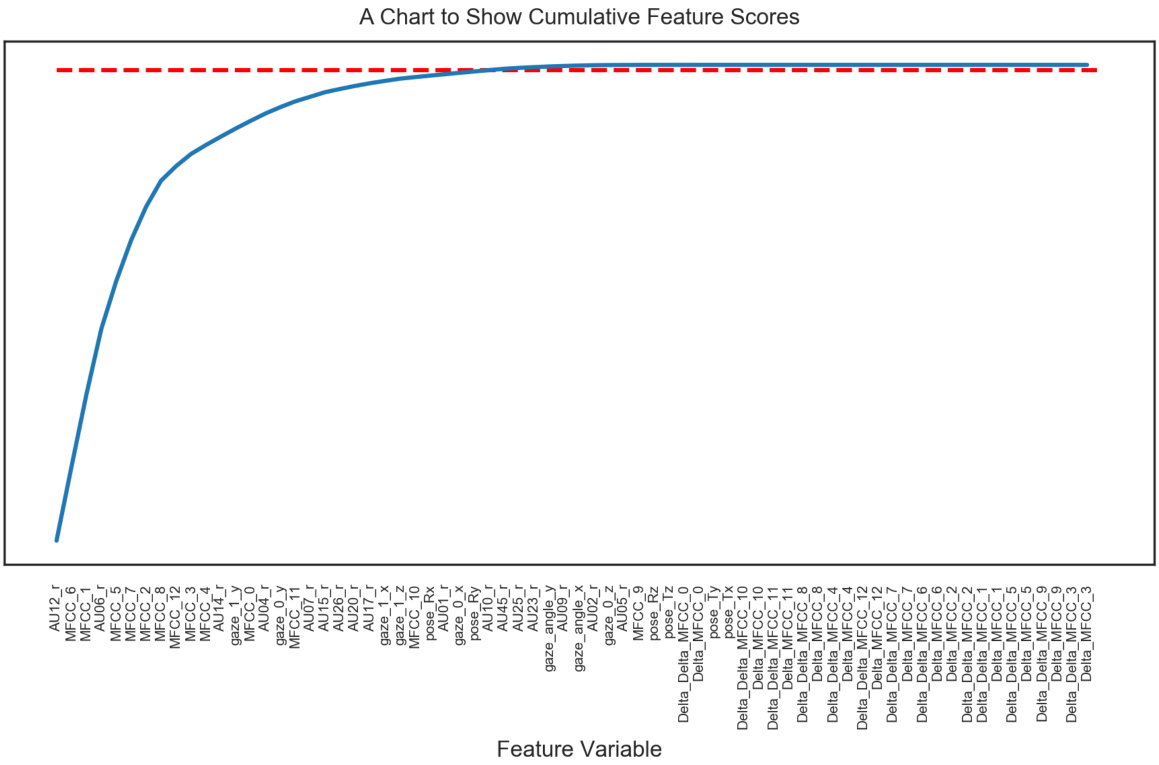

Figure 6.9: A chart to show cumulative feature scores of our 70 chosen features. The red line represents 99% of the cumulative feature scores, features to the right of this intersection have very low chi-squared scores and hence offer little information towards classification.

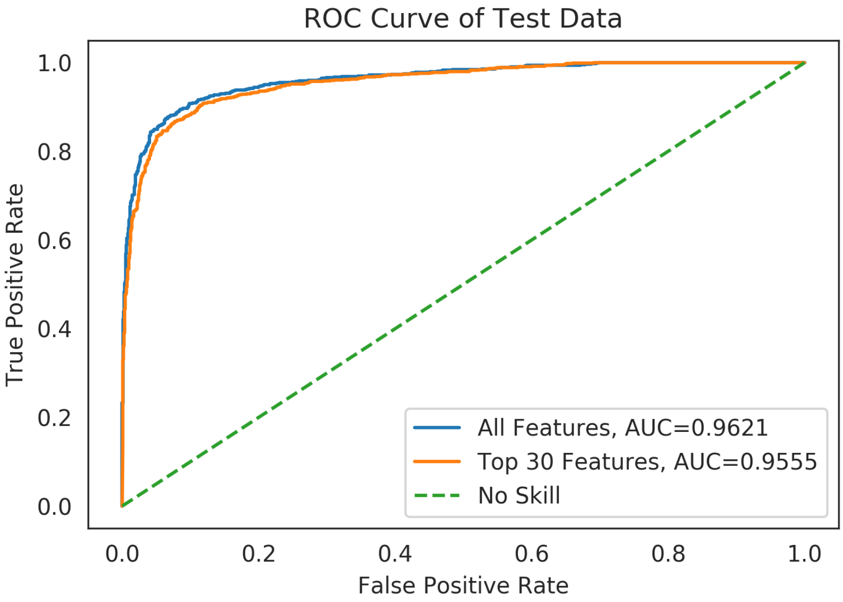

Figure 6.10: A ROC curve of the modified models performance on the test dataset, trained on the 30 top scoring features.

space compared to our original ones can be seen in Figure 6.10. We observe very little difference in the ROC graphs between the two methods, yet training the network of top features took just over 1/3 of the time it took to train the original network. Therefore, if we wished to expand the number of input videos in our training data, say the whole DFDC dataset, to reduce the number of GPU hours this would take without taking a significant performance hit, we could use these 30 features.

The top 30 features contain all but one of the MFCC features, yet no Δ or $\Delta\Delta$ MFCCs, showing that perhaps all the information needed to analyse audio signals is included in the MFCC, and the differentials are just adding redundancy. Furthermore, there are no pose location features, only pose rotation in the x,y planes in the top 30. This likely occurs since the DFDC has a large diversity of videos that distance from the camera varies across all, both real and fake, so it is not useful towards its classification. At the same time, head rotation is likely useful due to the nature of deepfake generation - to overlay the imposters face it has undergone rotation, stretch, or some other distortion.

The feature with the highest chi-squared score, and hence the 'most important', was AU12, the intensity of the lip corner puller. We hypothesise that this is the most important since some deepfake generation algorithms are not able to capture the subtle mouth movements correctly and often leave artifacts in the mouth region. Finally, we highlight that AU45, the intensity of blink, is not in these top features, despite the very first papers ([59]) in this area being able to detect deepfakes solely on its unnatural blinking pattern. Eye blinking intensity is no longer enough to classify deepfakes since, as previously mentioned, recent deepfake algorithms have now incorporated blinking into their generation algorithms.

Overall, we have provided a reduced feature space that achieves a similar performance to our original model. We identified that for audio, only the MFCCs (and not their Δ or $\Delta\Delta$) are useful. At the same time, for the video stream, only a subset of 18 of our original 30 visual features, specifically those relating to the lip area, as well as gaze and head rotation, all contain important information to allow for classification. As a result, we achieved only a 0.66 AUC performance decrease, yet a significnalty faster training time.

6.6 Limitations

Despite our methods ability to outperform state of the art, we acknowledge that our work is not without limitations. Firstly, we have not considered instances where multiple people occur at once in the frame. While the OpenFace library, used to extract the facial features, can do so, we did not include this as a possibility when processing our data since our training data did not have any instances of multiple actors in a frame. Yet, to allow our method to be used in the real world, further work would need to extend the approach to allow multiple persons.

Secondly, we have not tested to see what happens if a video contains no audio. Although we hope that the performance would revert to similar results gained in the unimodal visual method, we cannot be sure as we do not know what high-level abstractions the network learnt on our training set. Therefore the performance of detection may drop if there is no audio present to process.

Lastly, we wish to highlight the potential for bias in our model and the libraries used. Many datasets in this area tend to use caucasian individuals with British accents. While the DFDC has actors from many races across the genders, all footage appears to be English speaking. Therefore, we do not know what performance we would achieve on non-English speaking individuals. However, we trust that our approach is less affected by this bias than those who map to discrete emotions, such as [10] and [70], since the emotion classification datasets also suffer bias.

6.7 Summary

In this chapter, we have reviewed both unimodal and multimodal approaches to deepfake detection and tested four different ways to combine the modalities for the multimodal approach. Ultimately, we defined our final model, a CNN-BiLSTM with early additive fusion, to achieve 96.2 AUC on our hold-out data. We also tested our network on a second dataset, DF-TIMIT, to achieve a near-perfect AUC of 99.9, and outperformed SOTA on both datasets. Whilst also discussing that some datasets are ‘easier’ than others. Finally, we highlighted some limitations of our model, as well as what features provide the most information for classification.

Chapter 7

Conclusion

This chapter summarises the work conducted, a reflection of the project, and potential areas for future work.

7.1 Work Completed

This project aimed to explore the link between emotion conveyed in physical aspects of an individual with the emotion conveyed in their speech to be able to classify a video as real or fake.

We began by exploring what audio and visual features are commonly used in affective computing, as well as current methods of deepfake detection, to aid the selection of our multimodal features. Then, using the DFDC dataset, we tested both unimodal and multimodal approaches, also exploring different fusion options (early, mid-layer and late) for our audio-visual modalities. From the results of these tests, we decided on our final model, whose input is the concatenation of the audio-visual features, and architecture consists of CNNs and Bi-LSTMs. This achieved 96.2 AUC, an increase of 5.6% on current state-of-the-art methods. We also show our models ability to generalise to other datasets as we achieved 99.9 AUC on the DF-TIMIT dataset. Ultimately, we showed that multimodal approaches outperform their unimodal counterparts, and by investigating the top features, we confirm that both the audio's MFCC features and visual features provide important classification information.

7.2 Lessons Learned

Aside from furthering skills in deep learning, research, and how to generate and detect deepfake footage, the biggest lesson came from the scale of resources needed to process video data. The DFDC dataset is very large, and attempting to download and process the whole dataset took a considerable amount of time. Whatsmore, attempting to train on the whole dataset requires more storage and RAM than any cloud computing service provides without contacting them to increase your quota. Therefore, a significant amount of time was lost at the beginning of the project before deciding it was best to perform training on a subset of the available data.

7.3 Future Work

Future work should extend the model to be able to perform detection if more than one person is present in the footage, which our current method cannot as we did not include any processing for that case, nor are there any examples of multiple people in a frame in our training set.

Furthermore, we would like to investigate incorporating other modalities, such as linguistic analysis or even contextual information, to expand on the audio-visual approach. We have shown that multimodal approaches outperform unimodal ones; therefore, the more sources of information we can incorporate, the likely better we can detect fake footage.

Lastly, while not a deep learning approach to tackling deepfakes, we propose that legislation and social media sites should do more to tackle the spread of deepfake footage online. Using AI to detect deepfakes will likely be a cat-and-mouse game for decades to come, as new content creation methods will be made to evade current detection approaches. Therefore, we look towards other methods of dealing with this technology, although we understand that implementing laws to tackle deepfakes will also have many challenges.

7.4 Final Remarks

This report has proposed a multimodal deep learning solution to detect deepfakes that outperforms state-of-the-art methods and shown it can generalise to new data. We explored the link between audio-visual modalities and their various fusion options, selecting features commonly used in emotion recognition to train a CNN Bi-LSTM based classifier using an early additive fusion of these modalities. Although we initially hypothesised that there would be a correlation between the audio-visual features, this was not the case as the strong correlations were within the same modality. However, we showed that while not directly correlated, the deep network will learn a joint representation of audio-visual features, highlighting the need for multimodal methods to deepfake detection. Ultimately, we are confident in our methods ability to detect deepfakes, achieving an AUC score of 96.2.

Bibliography

[1] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *CoRR*, vol. abs/2003.05991, 2020.

[2] S. Singh, R. Sharma, and A. F. Smeaton, "Using gans to synthesise minimum training data for deepfake generation," *arXiv preprint arXiv:2011.05421*, 2020.

[3] Q. J. Lei Zhang, Yan Tong, "Active image labeling and its application to facial action labeling," *European Conference on Computer Vision (ECCV)*, vol. 10, pp. 706–719, 2008.

[4] M. Westerlund, "The emergence of deepfake technology: A review," *Technology Innovation Management Review*, vol. 9, pp. 39–52, 11 2019.

[5] S. Ahmed, "Navigating the maze: Deepfakes, cognitive ability, and social media news skepticism," *New Media & Society*, p. 14614448211019198, 2021.

[6] P. Fraga-Lamas and T. M. Fernández-Caramés, "Fake news, disinformation, and deepfakes: Leveraging distributed ledger technologies and blockchain to combat digital deception and counterfeit reality," *IT Professional*, vol. 22, no. 2, pp. 53–59, 2020.

[7] M. Albahar and J. Almalki, "Deepfakes: Threats and countermeasures systematic review," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 22, pp. 3242–3250, 2019.

[8] N. Diakopoulos and D. Johnson, "Anticipating and addressing the ethical implications of deepfakes in the context of elections," *New Media & Society*, p. 1461444820925811, 2019.

[9] L. Verdoliva, "Media forensics and deepfakes: an overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.

[10] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: An audio-visual deepfake detection method using affective cues," in *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, (New York, NY, USA), p. 2823–2832, Association for Computing Machinery, 2020.

[11] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, "Lips don't lie: A generalisable and robust approach to face forgery detection," 2021.

[12] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "M3er: Multiplicative multimodal emotion recognition using facial, textual, and speech cues," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 1359–1367, 2020.

[13] M. E. Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern recognition*, vol. 44, pp. 572–587, 2011.

[14] A. Mehrabian, "Communication without words," *Psychology Today*, vol. 2 (9), pp. 52–55, 1968.

[15] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Canton-Ferrer, "The deepfake detection challenge dataset," *CoRR*, vol. abs/2006.07397, 2020.

[16] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[17] D. Li, J. Liu, Z. Yang, L. Sun, and Z. Wang, "Speech emotion recognition using recurrent neural networks with directional self-attention," *Expert Systems with Applications*, vol. 173, p. 114683, 2021.

[18] K. M. M. Prabhu, *Window Functions and Their Applications in Signal Processing.* Baton Rouge: CRC Press, 2014.

[19] T. Giannakopoulos and A. Pikrakis, "Chapter 4 - audio features," in *Introduction to Audio Analysis*, pp. 59–103, Elsevier Ltd, 2014.

[20] S. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE transactions on neural networks*, vol. 3 5, pp. 683–97, 1992.

[21] M. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14, pp. 2627–2636, 1998.

[22] H. Abdi, "A neural network primer," *Journal of Biological Systems*, vol. 02, pp. 247–281, 1994.

[23] H. Ramchoun, M. Amine, M. A. Janati Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer perceptron: Architecture optimization and training," *International Journal of Interactive Multimedia and Artificial Inteligence*, vol. 4, pp. 26–30, 01 2016.

[24] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, *Fundamental Concepts of Convolutional Neural Network*, pp. 519–567. 01 2020.

[25] C. Zhou, C. Sun, Z. Liu, and F. Lau, “A c-lstm neural network for text classification,” *arXiv preprint arXiv:1511.08630*, 2015.

[26] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm networks,” in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4, pp. 2047–2052 vol. 4, 2005.

[27] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “Lstm: A search space odyssey,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[28] E. Kiperwasser and Y. Goldberg, “Simple and accurate dependency parsing using bidirectional LSTM feature representations,” *CoRR*, vol. abs/1603.04351, 2016.

[29] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005. IJCNN 2005.

[30] T. Shen, R. Liu, J. Bai, and Z. Li, ““deep fakes” using generative adversarial networks (gan),” 2018.

[31] M. Du, S. K. Pentyala, Y. Li, and X. Hu, “Towards generalizable forgery detection with locality-aware autoencoder,” *CoRR*, vol. abs/1909.05999, 2019.

[32] J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine Learning Refined: Foundations, Algorithms, and Applications.* Cambridge University Press, 2 ed., 2020.

[33] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.

[34] T. Fawcett, “Introduction to roc analysis,” *Pattern Recognition Letters*, vol. 27, pp. 861–874, 06 2006.

[35] J. Huang and C. Ling, “Using auc and accuracy in evaluating learning algorithms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[36] A. Deshmukh and S. B. Wankhade, “Deepfake detection approaches using deep learning: A systematic review,” *Intelligent Computing and Networking*, pp. 293–302, 2021.

[37] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, “Deep learning for deepfakes creation and detection,” *CoRR*, vol. abs/1909.11573, 2019.

[38] S. Lyu, "Deepfake detection: Current challenges and next steps," in *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 1–6, 2020.

[39] R. W. Picard, *Affective computing.* Cambridge, MA: MIT Press, 1997.

[40] X. Zhao and S. Zhang, "A review on facial expression recognition: Feature extraction and classification," *Technical review - IETE*, vol. 33, no. 5, pp. 505–517, 2016.

[41] L. Zhang and D. Tjondronegoro, "Facial expression recognition using facial movement features," *IEEE Transactions on Affective Computing*, vol. 2, no. 4, pp. 219–229, 2011.

[42] J. Chen, Z. Chen, Z. Chi, and H. Fu, "Facial expression recognition in video with multiple feature fusion," *IEEE Transactions on Affective Computing*, vol. 9, no. 1, pp. 38–50, 2018.

[43] K.-E. Ko and K.-B. Sim, "Development of a facial emotion recognition method based on combining aam with dbn," in *2010 International Conference on Cyberworlds*, pp. 87–91, 2010.

[44] J.-H. Kim, B.-G. Kim, P. P. Roy, and D.-M. Jeong, "Efficient facial expression recognition algorithm based on hierarchical deep neural network structure," *IEEE access*, vol. 7, pp. 41273–41285, 2019.

[45] A. Majumder, L. Behera, and V. K. Subramanian, "Emotion recognition from geometric facial features using self-organizing map," *Pattern Recognition*, vol. 47, no. 3, pp. 1282–1293, 2014. Handwriting Recognition and other PR Applications.

[46] S. Anwar, M. Milanova, S. Anwar, and Z. Svetleff, *Emotion Recognition and Eye Gaze Estimation System: EREGE*, pp. 364–371. 06 2018.

[47] S. Wu, Z. Du, W. Li, D. Huang, and Y. Wang, "Continuous emotion recognition in videos by fusing facial expression, head pose and eye gaze," (New York, NY, USA), Association for Computing Machinery, 2019.

[48] R. Jahangir, Y. W. Teh, H. F. Nweke, G. Mujtaba, M. A. Al-Garadi, and I. Ali, "Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges," *Expert Systems with Applications*, vol. 171, p. 114591, 2021.

[49] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: A review," *Int. J. Speech Technol.*, vol. 15, p. 99–117, June 2012.

[50] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2227–2231, 2017.

[51] I. Almajai, B. Milner, and J. Darch, "Analysis of correlation between audio and visual speech features for clean audio feature prediction in noise," in *Ninth International Conference on Spoken Language Processing*, 2006.

[52] R. Tolosana, S. Romero-Tapiador, J. Fiérrez, and R. Vera-Rodríguez, "Deepfakes evolution: Analysis of facial regions and fake detection performance," *CoRR*, vol. abs/2004.07532, 2020.

[53] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *CoRR*, vol. abs/1812.08685, 2018.

[54] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," *CoRR*, vol. abs/1901.08971, 2019.

[55] B. Zi, M. Chang, J. Chen, X. Ma, and Y. Jiang, "Wilddeepfake: A challenging real-world dataset for deepfake detection," *CoRR*, vol. abs/2101.01456, 2021.

[56] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A new dataset for deepfake forensics," *CoRR*, vol. abs/1909.12962, 2019.

[57] F. AI, "Deepfake detection challenge results: An open initiative to advance ai." `https://ai.facebook.com/blog/deepfake-detection-challenge-results-an-open-initiative-to-advance-ai/`, 2020. [Online; accessed 6-July-2021].

[58] N. Dufour and A. Gully, "Contributing data to deepfake detection research," Sep 2019. Last accessed on 16/08/21.

[59] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai created fake videos by detecting eye blinking," pp. 1–7, 12 2018.

[60] W. Jiang and R. Yasrab, "Fighting deepfakes using body language analysis," 01 2021.

[61] I. Demir and U. A. Ciftci, "Where do deep fakes look? synthetic face detection via gaze tracking," *CoRR*, vol. abs/2101.01165, 2021.

[62] U. A. Ciftci, I. Demir, and L. Yin, "How do the hearts of deep fakes beat? deep fake source detection via interpreting residuals with biological signals," *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–10, 2020.

[63] S. Agarwal, H. Farid, O. Fried, and M. Agrawala, "Detecting deep-fake videos from phoneme-viseme mismatches," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2814–2822, 2020.

[64] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," 2018.

[65] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting World Leaders Against Deep Fakes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, (Long Beach, CA), p. 8, IEEE, June 2019.

[66] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of audio deepfake detection," in *Odyssey*, 2020.

[67] M. Shan and T. Tsai, "A cross-verification approach for protecting world leaders from fake and tampered audio," *ArXiv*, vol. abs/2010.12173, 2020.

[68] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," in *INTERSPEECH*, 2015.

[69] A. Lieto, D. Moro, F. Devoti, C. Parera, V. Lipari, P. Bestagini, and S. Tubaro, ""hello? who am i talking to?" a shallow cnn approach for human vs. bot speech classification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2577–2581, 2019.

[70] K. Chugh, P. Gupta, A. Dhall, and R. Subramanian, "Not made for each other-audio-visual dissonance-based deepfake detection and localization," in *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, (New York, NY, USA), p. 439–447, Association for Computing Machinery, 2020.

[71] Y. Gu, X. Zhao, C. Gong, and X. Yi, "Deepfake video detection using audio-visual consistency," in *Digital Forensics and Watermarking* (X. Zhao, Y.-Q. Shi, A. Piva, and H. J. Kim, eds.), (Cham), pp. 168–180, Springer International Publishing, 2021.

[72] M. Lomnitz, Z. Hampel-Arias, V. Sandesara, and S. Hu, "Multimodal approach for deepfake detection," in *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–9, 2020.

[73] H. Gunes and M. Piccardi, "Bi-modal emotion recognition from expressive face and body gestures," *Journal of Network and Computer Applications*, vol. 30, pp. 1334–1345, 11 2007.

[74] T. Baltruaitis, A. Zadeh, Y. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 59–66, 2018.

[75] K. Liu, Y. Li, N. Xu, and P. Natarajan, "Learn to combine modalities in multimodal deep learning," 2018.

[76] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L. Morency, "Tensor fusion network for multimodal sentiment analysis," *CoRR*, vol. abs/1707.07250, 2017.

[77] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *Advances in Biometrics* (M. Tistarelli and M. S. Nixon, eds.), (Berlin, Heidelberg), pp. 199–208, Springer Berlin Heidelberg, 2009.

[78] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2307–2311, 2019.

[79] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–8, 2019.

[80] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8261–8265, 2019.

[81] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1831–1839, 2017.

[82] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 83–92, 2019.

[83] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: a compact facial video forgery detection network," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, 2018.

[84] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "Faceforensics++: Learning to detect manipulated facial images," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–11, 2019.

[85] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," *CoRR*, vol. abs/1811.00656, 2018.

[86] B. Xu, J. Liu, J. Liang, Z. Wei, and Y. Zhang, "Deepfake videos detection based on texture features," *Computers, Materials & Continua*, vol. 680, pp. 1375–1388, 01 2021.

Appendix A

Full Results

We provide precision and recall results:

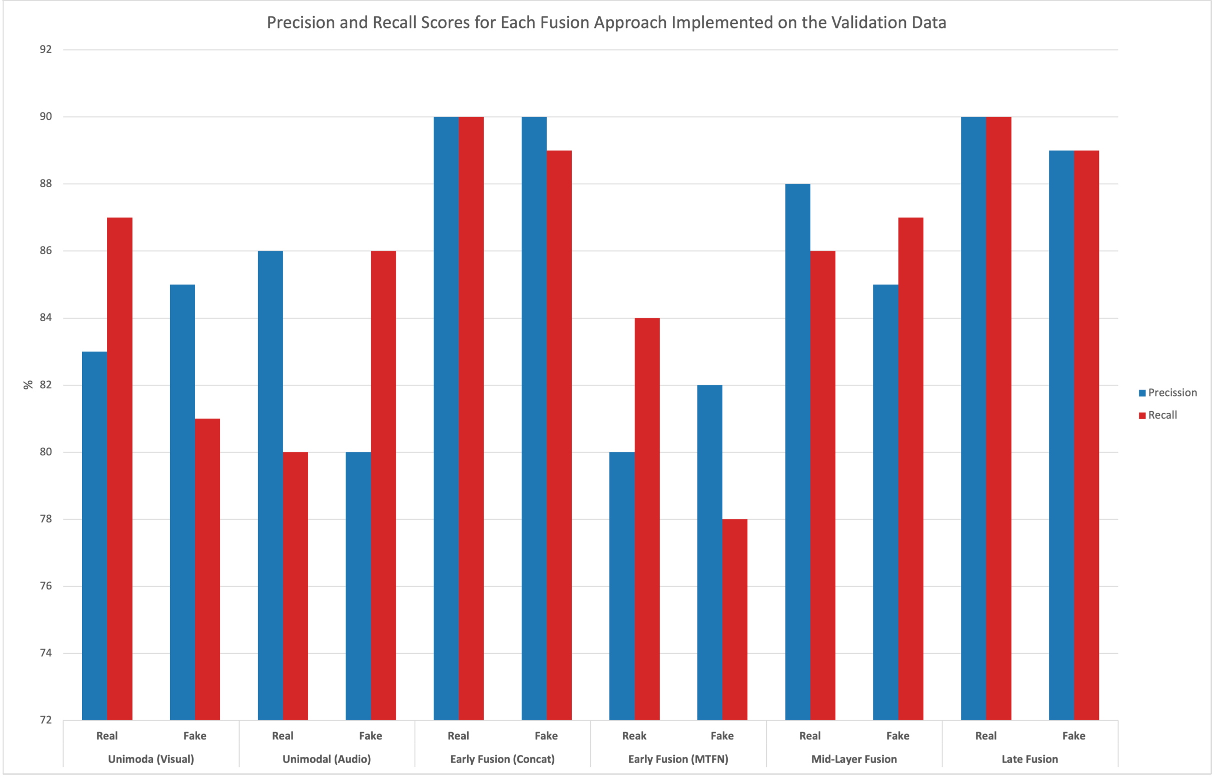

Figure A.1: Precision and recall scores for the different modalities and fusion methods on the validation data.

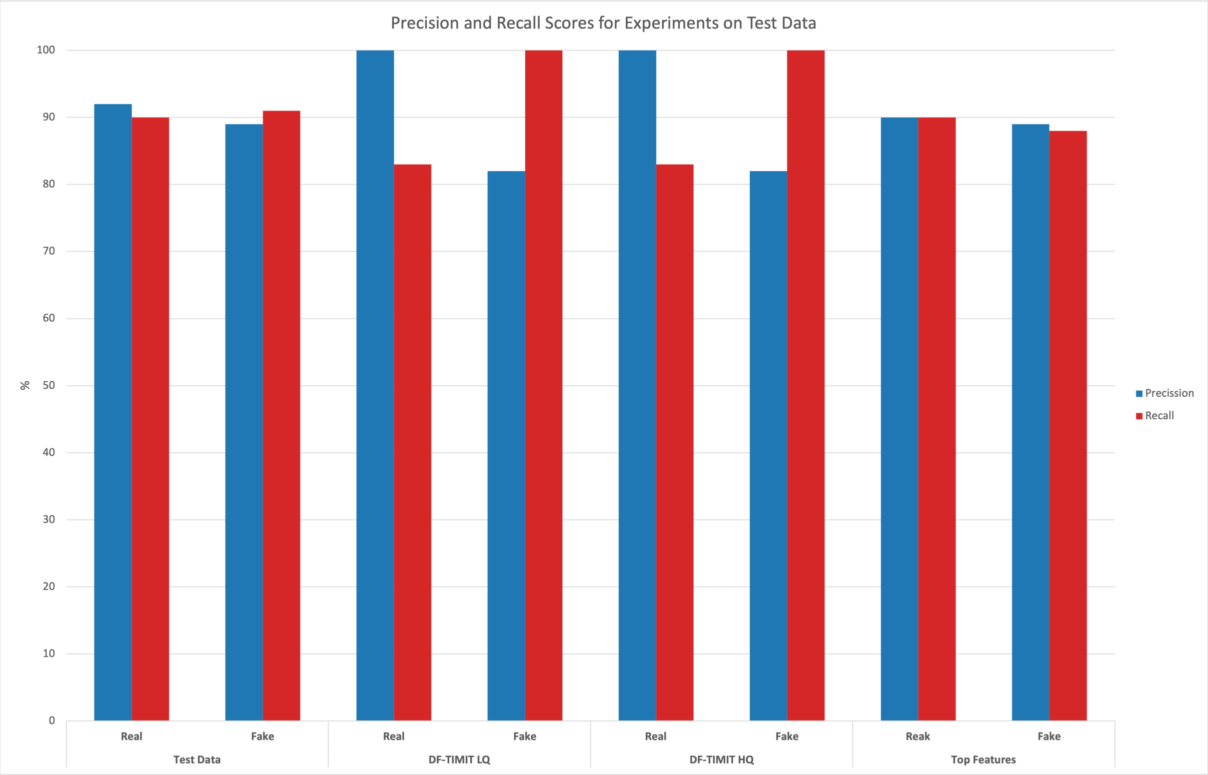

Figure A.2: Precision and recall scores for the final model on different test datasets.